



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**MULTIDIMENZIONÁLNÍ ANALÝZA DAT A ZPRACOVÁNÍ
ANALYTICKÉHO ZOBRAZENÍ**

MULTIDIMENSIONAL DATA ANALYSIS AND ANALYTIC VIEW PROCESSING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Veronika Foltýnová

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Vladislav Škorpil, CSc.

BRNO 2018

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Studentka: Bc. Veronika Foltýnová

ID: 164828

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Multidimenzionální analýza dat a zpracování analytického zobrazení

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s možnostmi návrhu a programování v jazyce C#. V teoretické části nastudujte problematiku a popište možnosti zobrazení multidimenzionálních dat SQL databázového systému. V praktické části navrhnete a implementujte strukturu aplikace pro možnost vyčítat a zobrazit data z SQL databáze. Zejména se tato problematika týká analytického zobrazení dat pasivní optické sítě.

DOPORUČENÁ LITERATURA:

[1] LACKO, L. Mistrovství v SQL Server 2012: [kompletní průvodce databázového experta]. Brno: Computer Press, 2013. ISBN 978-80-2513-773-4.

[2] PILGRIM, M. Ponořme se do Python(u) 3. CZ.NIC, z.s.p.o., 2010, ISBN: 978-80-904248-2-1

Termín zadání: 5.2.2018

Termín odevzdání: 21.5.2018

Vedoucí práce: doc. Ing. Vladislav Škorpil, CSc.

Konzultant: Ing. Václav Oujezský, Ing. Michal Jurčík

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato práce se zabývá problematikou analýzy a zobrazení multidimenzionálních dat. V teoretické části je uvedena problematika dolování dat, její úlohy a techniky a stručné vysvětlení pojmů Business Intelligence a datový sklad. Dále je nastíněna problematika databází. Následně jsou uvedeny možnosti zobrazení multidimenzionálních dat. Na závěr teoretické části je stručně vysvětlena problematika optických sítí, a zejména pojmy Gigabitová pasivní optická síť a její rámec, jelikož data z rámců této sítě budou zobrazována výslednou aplikací. V praktické části lze nalézt tvorbu zdrojové databáze a aplikaci pro vytvoření OLAP krychle a zobrazení multidimenzionálních dat v ní obsažených. Tvorba této aplikace vychází z teoretických poznatků multidimenzionálních databází a technologie OLAP.

Klíčová slova

databáze, dolování dat, GPON, multidimenzionální data, OLAP

Abstract

This thesis deals with the analysis and display of multidimensional data. In the theoretical part, the issue of data mining, its tasks and techniques, and a brief explanation of the terms Business Intelligence and data warehouse are presented. The issue of databases is also described in this thesis. Subsequently, the options for displaying multidimensional data are described. At the end of the theoretical part is briefly explained the problems of optical networks and especially the terms Gigabit passive optical network and its frame, because the data from the frames of this network will be displayed by an application.

In the practical part, you can find creating a source database and an application to create a OLAP cube and display multidimensional data. This application is based on the theoretical knowledge of multidimensional databases and OLAP technology.

Keywords

database, data mining, GPON, multidimensional data, OLAP

Bibliografická citace:

FOLTÝNOVÁ, V. Multidimenzionální analýza dat a zpracování analytického zobrazení. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 61 s. Vedoucí diplomové práce doc. Ing. Vladislav Škorpil, CSc.

Prohlášení

„Prohlašuji, že svou závěrečnou práci na téma Multidimenzionální analýza dat a zpracování analytického zobrazení jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....
podpis autora

Výzkum popsáný v této diplomové práci byl realizován v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Vladislavu Škorpilovi, CSc. a konzultantovi diplomové práce Ing. Václavu Oujezskému, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce. Dále děkuji za spolupráci kolegovi Bc. Martinu Holíkovi.

V Brně dne

.....
podpis autora(-ky)

Obsah

1	Úvod	11
2	Dolování dat	12
2.1	Úlohy dolování dat	12
2.2	Techniky dolování dat	13
2.2.1	Statistické metody	13
2.2.2	Umělé neuronové sítě	15
2.2.3	Genetické algoritmy	16
2.3	Business Intelligence (BI)	17
2.4	Datový sklad (Data Warehouse)	17
3	Databáze	18
3.1	Databáze – databázový systém	18
3.2	Databázové modely dat	19
3.3	Relační databáze	22
3.3.1	Tabulky relační databáze	22
3.4	Dotazovací jazyky	24
3.5	Multidimenzionální databáze	25
3.5.1	Multidimenzionální databázový model	25
3.5.2	Dotazovací jazyk MDX (Multidimensional Expressions)	26
3.5.3	Vytvoření OLAP krychle z relační databáze	27
4	Zobrazení multidimenzionálních dat	38
4.1	Grafické zobrazení multidimenzionálních dat	38
5	Pasivní optické sítě (PON)	42
6	Tvorba Zdrojové databáze s daty GPON	45
7	Návrh aplikace pro analytické zobrazení multidimenzionálních dat	50
7.1	Aplikace pro zobrazení multidimenzionálních dat	50
7.1.1	Tvorba aplikace	50
8	Závěr	56
	Literatura	57
	Seznam symbolů, veličin a zkratk	60
	Seznam příloh	61

Seznam obrázků

Obr. 1.1: Obecný graf lineární regrese	14
Obr. 1.2: Struktura neuronu (perceptoru)	15
Obr. 1.3: Struktura neuronové sítě.....	16
Obr. 2.1: Ukázka hierarchického modelu dat	19
Obr. 2.2: Ukázka síťového modelu dat.....	20
Obr. 2.3: Ukázka Entity Relationship modelu	21
Obr. 2.3: Ukázka objektového datového modelu	21
Obr. 2.4: Návrh tabulky relační databáze	22
Obr. 2.5: Tabulka relační databáze s daty.....	22
Obr. 2.6: Ukázka vztahu 1:1.....	23
Obr. 2.7: Ukázka vztahu 1:N.....	23
Obr. 2.8: Ukázka vztahu N:M.....	24
Obr. 2.9: Multidimenzionální krychle.....	25
Obr. 2.10: Multidimenzionální krychle: a) Analýza údajů za časové období, b) Analýza údajů dle státu, c) Analýza údajů dle výrobku nebo skupiny výrobků	26
Programový kód 1: Struktura příkazu SELECT jazyka MDX	26
Obr. 2.11: Obnovení databáze	28
Obr. 2.12: Výběr souboru zálohy databáze	28
Obr. 2.13: Připojování zdroje dat.....	30
Obr. 2.14: Připojení databáze jako zdroje dat.....	31
Obr. 2.15: Výběr tabulek do pohledu (Data Source View)	32
Obr. 2.16: Vlastnosti tabulky v pohledu (Data Source View)	32
Obr. 2.17: Přejmenování tabulek v pohledu (Data Source View).....	33
Obr. 2.18: Pohled strukturu vytvořené krychle OLAP (Cubes)	34
Obr. 2.19: Úprava vlastností dimenzí: a) dimenze Customers, b) dimenze Products	35
Obr. 2.20: Potvrzení nasazení OLAP kostky.....	36
Obr. 2.22: Úspěšné zpracování dat multidimenzionální databáze.....	37
Obr. 3.1: Sloupcový graf zobrazující jednu dimenzi a jeden fakt.....	39
Obr. 3.2: Pruhový (Sloupcový) graf zobrazující dvě dimenze a jeden fakt	39
Obr. 3.3: Spojnicový graf zobrazující dvě dimenze a jeden fakt.....	40
Obr. 3.4: Výsečový graf zobrazující jednu dimenzi a jeden fakt.....	41
Obr. 3.5: Graf XY zobrazující jednu dimenzi a dva fakty	41
Obr. 4.1: Blokové schéma OAN.....	42
Obr. 4.2: Diagram rámce pro downstream.....	43
Obr. 4.3: Diagram rámce pro upstream	44
Obr. 5.1: Nastavení databáze (General).....	46

Obr. 5.2: Nastavení databáze (Options).....	46
Obr. 5.3: Schéma tabulek pro GPON.....	47
Obr. 5.4: Schéma tabulky PCBDheader	49

Seznam programových kódů

Programový kód 1: Struktura příkazu SELECT jazyka MDX.....	26
--	----

1 ÚVOD

V dnešní době, vlivem rychlého technologického rozvoje, nabývá čím dál více na důležitosti komodita, s níž se každý člověk setkává den co den v její hmatatelné či abstraktní podobě, a ač o této skutečnosti nemusí vědět, je na ní jistým způsobem závislý. Touto komoditou jsou informace. Informace provázejí člověka od nepaměti a vždy byly důležitou součástí lidského vývoje, ať v oblasti vědy, techniky nebo obchodu. V dřívějších dobách lidé neměli k dispozici takové množství informací, jako máme dnes, ale zároveň k jeho zpracování měli pouze omezené prostředky, tedy vlastní zkušenosti a logické myšlení. Tak velké množství informací, jaké dokážeme v současné době z různých zdrojů získat, ale nedokáže lidská mysl uchovávat v takové míře a efektivitě, jako to dokáží současné technologická řešení.

Velké množství informací není přínosné, pokud je nelze efektivně zpracovat. Potřebujeme tedy informace získat, uchovat z nich získaná data, analyzovat a vhodně tato data prezentovat. Nejdůležitější součástí celého uvedeného procesu je způsob uložení dat. Každá operace využívá jiného typu úložiště. Pro sběr dat jsou využívány databáze. Jelikož při sběru dat může dojít k jejich chybnému či neúplnému vložení, je nutné data předzpracovat a uložit v datovém skladu. Datový sklad je v současné době využíván pro ukládání dat, která jsou předzpracována a lze na nich provádět potřebné analýzy tak, aby byly získány požadované informace. Předzpracování dat lze provést pomocí dolování dat, případně dobývání znalostí z databází. Pro lepší přístup a snazší práci s těmito daty je využívána technologie OLAP (Online Analytical Processing) a multidimenzionálního přístupu k uloženým datům.

Cílem této práce je prostudovat možnosti analýzy a zobrazení multidimenzionálních dat, návrh a vytvoření aplikace pro analytické zobrazení dat z GPON rámců. Pro uchovávání dat, jež budou analyzována, je jako databáze použit MS SQL Server. Programovacím jazykem pro samotnou aplikaci je zvolen C#.

Tato práce je členěna do sedmi částí. V kapitole 2 je vysvětlena problematika dolování dat a jeho metody. Kapitola 3 obsahuje seznámení s databázemi. Kapitola 4 nabízí možnosti zobrazení multidimenzionálních dat. V 5. kapitole je stručně vysvětlena problematika pasivních optických sítí. Následující kapitola se zaměřuje na vytvoření zdrojové databáze. Na samotný návrh aplikace je zaměřena kapitola 7. Kapitola 8 představuje shrnutí celé práce.

2 DOLOVÁNÍ DAT

Pojem dolování dat označuje proces hledání znalostí a informací ve velkém objemu dat. Pomocí dolování dat hledáme skryté souvislosti mezi daty. Pojem dolování dat (data mining) je často srovnáván s pojmem dobývání znalostí z databází (Knowledge discovery in databases, KDD). V literatuře jsou často používána jako synonyma. Oba tyto pojmy znamenají totéž, jen s tím rozdílem, že u dobývání znalostí z databází je důležitá i samotná příprava dat.

Analogii obecného postupu dolování dat je možné najít u dolování drahých kovů. V první fázi na základě neucelených informací a odhadů jen předpokládáme, že se v dané oblasti nacházejí požadovaná data (drahý kov). Tuto skutečnost musíme ověřit na testovacích vzorcích. V tomto případě je určeno, zda je naše hypotéza správná či nikoli. Pokud ano, začneme těžit data (horninu). Po vytěžení je nutné získanou surovinu očistit. A následují další vyhodnocovací a zhodnocovací procesy.

Na první pohled by se mohlo zdát, že jde o univerzální a všemocnou metodu získávání dat. Ovšem není tomu tak. Dolování dat je náročná a také v jistých případech náhodná metoda. Při jejím provádění se používají heuristické algoritmy, neuronové sítě a jiné programové technologie a metody umělé inteligence. Pro svoji náročnost se začalo dolování dat prosazovat až v posledních letech, kdy rozvoj tímto směrem umožnil pokrok vědy a techniky.

Jelikož o této problematice bylo sepsáno mnoho knih, budou zde shrnuta nejdůležitější fakta a používané úlohy a techniky. Při větším zájmu o tuto problematiku je možné čerpat z uvedené literatury [1], [2].

2.1 Úlohy dolování dat

Dolování dat je možné použít v mnoha případech, například ke zjištění nejčastěji prodávaných komodit za určitý časový úsek nebo vyhodnocení ztrátovosti či výdělečnosti zakázek. Dle povahy řešených případů je možné rozčlenit jejich řešení do několika typů úloh uvedených níže.

Prediktivní

Cílem této úlohy je předpovězení hodnoty určité veličiny na základě hodnot ostatních veličin. V případě statistiky je takovouto metodou regresní analýza. Popis lineární regrese lze nalézt níže v podkapitole statistické metody (podkapitola č. 2.2.1).

Deskriptivní

Zde je podstatou popis celé datové množiny. Takovouto metodou může být analýza shluků. Při této analýze dohází k vytvoření skupin (shluků), do kterých je možné rozdělit projevy v datech, a to na základě podobnosti.

Explorační analýzy

Cílem této úlohy je zkoumání dat bez předcházející znalosti, která by směřovala hledání dat.

Hledání dle vzorů

V tomto případě máme k dispozici vzor a cílem je nalézt shodující se nebo podobné vzory ve zkoumaných datech. Realizace této úlohy se využívá při rozpoznávání obrázků a textů.

2.2 Techniky dolování dat

Jednotlivé úlohy dolování dat jsou řešeny pomocí různých technik. Níže jsou uvedeny nejdůležitější z nich.

Statistika obecně

V oblasti dolování dat je využíván statistický model. Jde o soubor matematických funkcí popisující chování objektů v cílové skupině z hlediska náhodných proměnných a jejich rozdělení pravděpodobnosti. Statistické modely jsou využívány zejména pro modelování dat a datových tříd.

2.2.1 Statistické metody

Korelace

Korelace je úroveň závislosti mezi dvěma proměnnými. Může být pozitivní i negativní. Pozitivní korelací je myšlen stav kdy vysokou úroveň jedné proměnné provází vysoká úroveň druhé proměnné. Negativní korelace naopak udává stav, při kterém vysokou úroveň jedné proměnné provází nízká úroveň druhé proměnné. Příkladem pozitivní korelace může být častý nákup dvou druhů zboží současně ve větší míře.

Lineární regrese

Jde o statistickou metodu regresní analýzy. Touto metodou zkoumáme závislost proměnných, jejichž hodnoty jsou získány při realizaci experimentů, měřeních apod. Charakter těchto proměnných je náhodný. Proměnné mohou být závislé či nezávislé. Pro reprezentaci nezávislých proměnných použijeme náhodný vektor X a pro reprezentaci závislých proměnných náhodnou veličinu Y . Závislost mezi X a Y vyjadřuje regresní funkce jež má obecný tvar:

$$y = (x, \beta) = (Y|X = x) \quad (1.1)$$

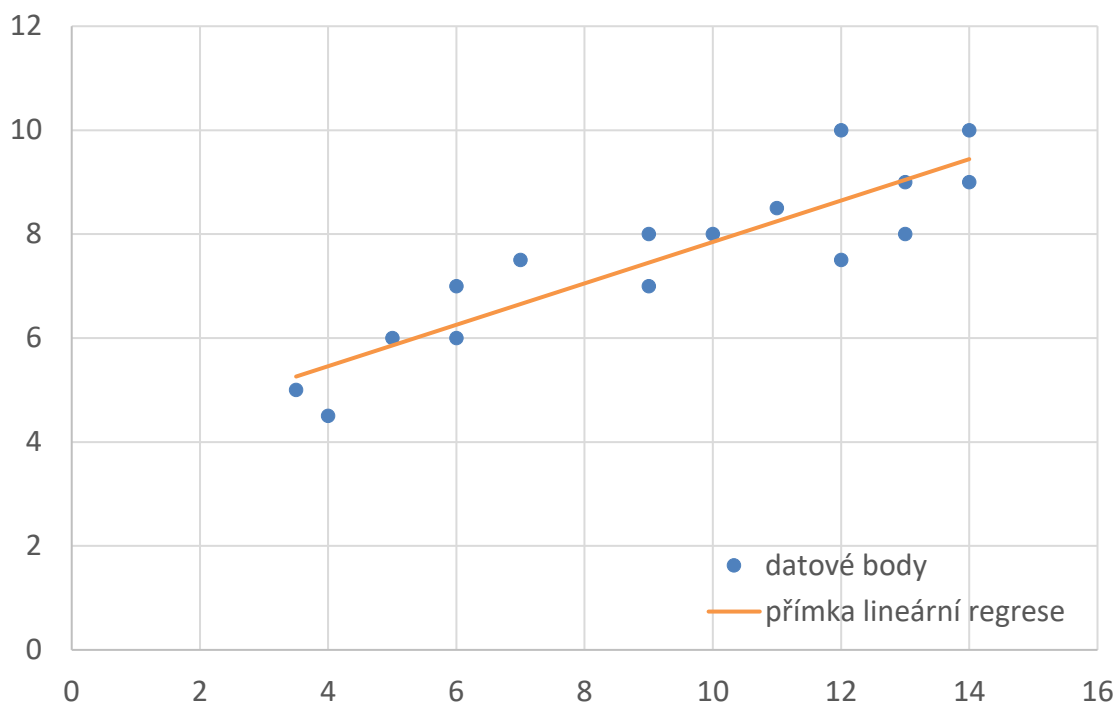
kde $x = (x_1, \dots, x_k)$ je vektor nezávislých proměnných (vektor X), y je závisle proměnná (hodnota veličiny Y), $\beta = (\beta_1, \dots, \beta_m)$ je vektor parametrů, tzv. regresních koeficientů β_j , $j = (1, \dots, m)$, a $E(Y|X = x)$ je podmíněná střední hodnota.

Jde-li o lineární regresi, má regresní funkce tvar:

$$y = \sum_{j=1}^m \beta_j f_j(x) \quad (1.2)$$

kde $f_j(x)$ jsou známé funkce neobsahující regresní koeficienty β_1, \dots, β_m .

Jednou z používaných metod lineární regrese je metoda nejmenších čtverců. Cílem této metody je najít přímku, která prochází mezi jednotlivými body (proměnnými), pro niž platí, že součet druhých mocnin odchylek od každého bodu je minimální. Příklad proložení bodů takovouto přímkou je možno vidět na obr. 1.1. Více o této metodě a obecně o lineární regresní analýze nalezneme v literatuře [3], [4].



Obr. 1.1: Obecný graf lineární regrese

Logistická regrese

Logistická regrese je stejně jako lineární regrese metodou regresní analýzy. Hodnota závislé proměnné Y má binární charakter. Nabývá tedy hodnot jedna, pokud nastal sledovaný jev a hodnoty nula, jestliže sledovaný jev nenastal. Jako u každého regresního modelu i zde je důležitý vztah mezi hodnotami X a Y , tedy regresní funkce. V tomto případě jde o vztah pravděpodobností úspěchu či neúspěchu k nezávislým hodnotám náhodného vektoru X , jak je uvedeno v (1.3) a (1.4).

$$\log it(p) = \ln\left(\frac{p}{1-p}\right) \quad (1.3)$$

$$\ln\left(\frac{p}{1-p}\right) = x^T \beta \quad (1.4)$$

Regresní funkce logistické regrese (1.4) je tvořena funkcí logit (1.3), jež je opakem logistické funkce tzv. sigmoidy, hodnoty náhodného vektoru X $x^T = (1, x_1, x_2, \dots, x_s)$ a parametr regresního modelu β .

Příklad využití lineární regrese lze nalézt v literatuře [5].

Předpovědi trendů

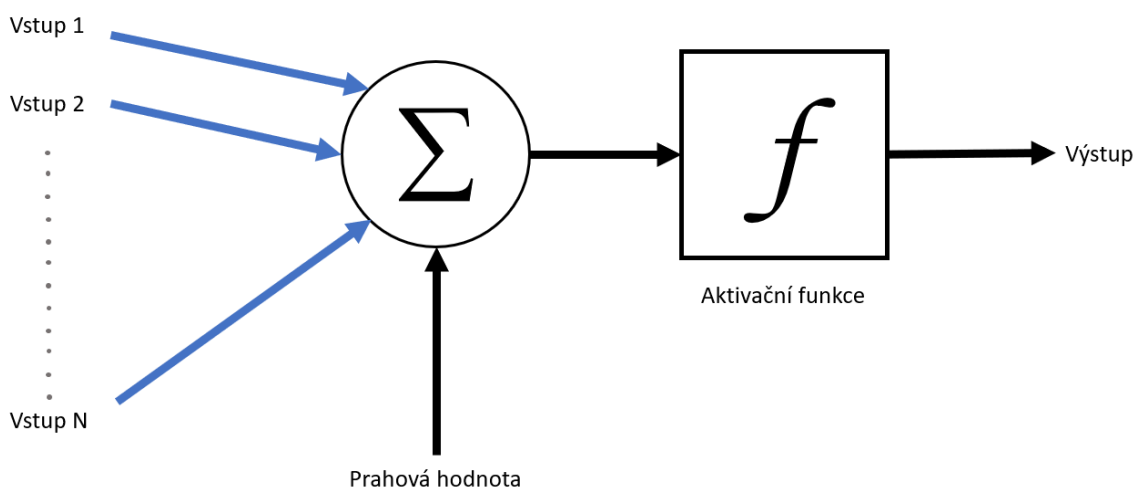
Trend je předpovídán pro určitou proměnnou. Tato předpověď je založena na analýze získaných údajů, na jejichž základě definujeme pravidla, s jejichž pomocí předpovídáme budoucí trend dané proměnné.

Pro předpověď trendů je možné použít neuronové sítě nebo regresi časových úseků.

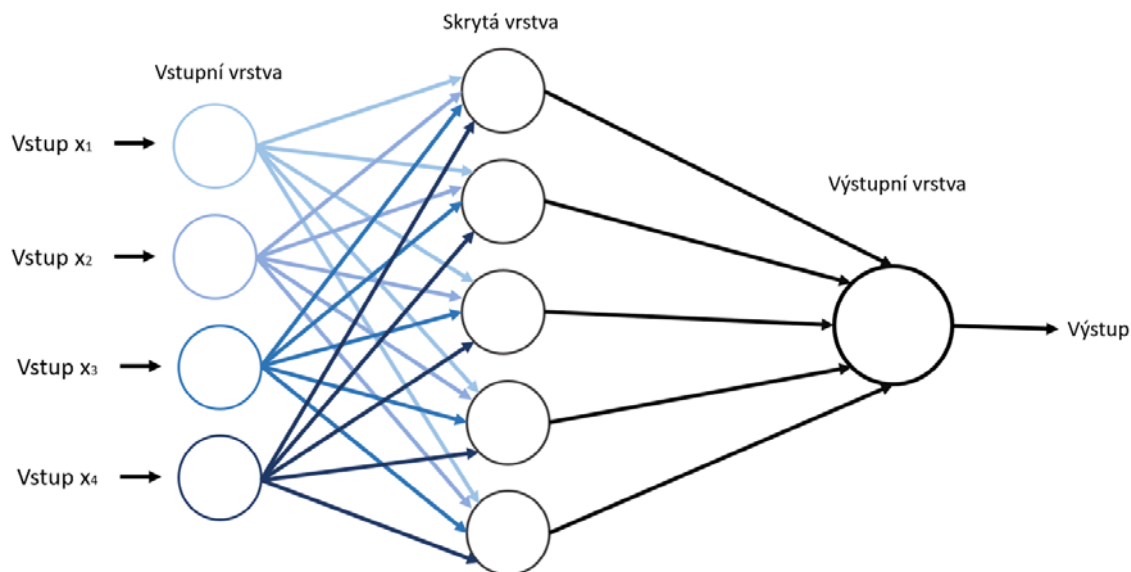
2.2.2 Umělé neuronové sítě

Umělé neuronové sítě jsou modely, jejichž struktura je inspirována lidským mozkem, tedy biologickou neuronovou sítí. Lidský mozek se při svém vývoji neustále učí. Tento vzor převzala i jeho uměle vytvořená podoba neuronové sítě. Metody, které využívají pro svoji funkci učení, jsou řazeny do skupiny metod strojového učení.

Neuronovou sítí je možné popsat jako orientovaný graf, který tvoří neurony a spoje mezi nimi. Základní stavební jednotkou neuronové sítě je neuron (perceptron). Lze jej popsat jako buňku do níž vstupuje více vstupních údajů buď z výstupu jiného neuronu nebo z vnějšího vstupu sítě. Neuron údaje zpracuje pomocí sumarizace, prahové hodnoty, což je potenciál neuronu a aktivační funkce. Dále údaje zpracované neuronem postupují do vstupu dalšího neuronu nebo na výstup neuronové sítě. Strukturu neuronu lze vidět níže na obr. 1.2 a strukturu jednoduché neuronové sítě lze vidět níže na obr. 1.3



Obr. 1.2: Struktura neuronu (perceptoru)



Obr. 1.3: Struktura neuronové sítě

Při větším zájmu o problematiku neuronových sítí je možno čerpat z literatury [6], [7].

2.2.3 Genetické algoritmy

Tyto algoritmy jsou postaveny na principu Darwinovy evoluční teorie [8] (teorie vzniku druhů). Jsou založeny na mechanismu přirozeného výběru, kdy přežije nejsilnější jedinec a na principech genetiky. Základní pojmy charakterizující genetické algoritmy jsou: chromozóm, gen, populace a fitness hodnota. Chromozóm je seskupení informací o vlastnostech a chování každého jedince. Jde o řetězec informací, kterým může být například: celé číslo, matice, vektory nebo křivky. Gen je nejmenší a zároveň dále nedělitelnou částí chromozómu. Populací nazýváme skupinu jedinců, která je popsána chromozómy v rámci jedné populace. Fitness hodnota je vyjádření kvality jedince pomocí číselné hodnoty. Obvykle jde o číslo v intervalu 0 až 1, ale může to být i číslo z jiného intervalu. Proto je nutné pro každý problém vytvořit fitness funkci, jejímž výstupem bude požadovaná číselná hodnota (fitness hodnota).

Na počátku řešení problému pomocí genetického algoritmu je vybrána náhodná populace jedinců z oblasti, v níž hledáme řešení. Každý jedinec této populace je charakterizován svou genetickou informací. Po výběru počáteční populace je nutné přidělit každému jedinci tzv. fitness hodnotu, jež znázorňuje kvalitu řešení (problému) reprezentovanu tímto jedincem. Dle této kvality jsou vybráni jedinci, kteří jsou dále upravováni. Tento výběr se nazývá selekce. Úpravami, pomocí nichž vzniká nová populace jsou křížení a mutace. Celý proces se opakuje do doby, než je dosaženo dostačující kvality, případně po předem stanovené době.

Při větším zájmu o problematiku genetických algoritmů je možno čerpat z literatury [8].

2.3 Business Intelligence (BI)

Jedná se o souhrn aplikačních řešení pro lepší orientaci ve velkém množství dat pro usnadnění a podporu rozhodovacích procesů ve firmě. Firmy v průběhu času sbírají velké množství dat a potřebují tato data přeměnit na užitečné informace. Tato přeměna dat je cílem Business Intelligence (BI). Příkladem přínosu BI je například zjištění, která komodita se nejlépe prodává, oblast, ve které firma vydělává více peněz nebo kde naopak prodělává. Na základě těchto informací může firma změnit svoji obchodní strategii a zajistit tak větší výdělků nebo lepší postavení na trhu. Ke splnění svých cílů využívá BI například proces dolování dat a datové sklady, viz následující kapitola.

Pojem Business Intelligence se většinou nepřekládá, uvádí se ve své dvouslovné podobě tak i ve dvoupísmenné zkratce BI.

Více informací o problematice Business Intelligence lze nalézt v literatuře [10].

2.4 Datový sklad (Data Warehouse)

Pojem datový sklad můžeme jednoduše definovat jako strukturované datové úložiště historických informací, využitelných k získávání informací a podporu rozhodování. Lze jej přirovnat ke skladům (depozitářům muzea). Exponáty a historické relikvie jsou zde uloženy a roztrženy například dle času, historického období apod.

Data jsou běžně uchovávána v operačních databázích. Z těchto databází jsou data sesbírána, předzpracována a uložena do datového skladu. Zde lze vykonávat analýzy potřebné pro rozhodování firmy.

Firma, jež potřebuje ke svému rozhodování určité informace, má tedy nejčastěji dvě možnosti, a to vybudování datového skladu svépomocí nebo nákup informací od zprostředkovatele, jenž potřebné informace poskytuje. Vybudování a provoz datového skladu je velice finančně náročné, a proto jej vlastní převážně společnosti s přístupem k velkému množství vstupních dat, což jsou například: banky, telekomunikační společnosti, mobilní operátoři, pojišťovny, velké obchodní řetězce apod.

3 DATABÁZE

Data jsou nedílnou součástí života a jsou důležitým faktorem pro uchovávání znalostí a informací, z čehož plyne potřeba tato data uchovávat. Nejdříve byla data uchovávána v papírových kartotékách. Ovšem toto řešení brzy začalo být nedostačující a při velkém objemu dat i časově náročné. Vyhledání potřebných dat trvalo obsluze takovéto papírové databáze velice dlouho.

První urychlení práce s daty přišlo s uchováváním dat na tzv. děrných štítcích. Informace se uchovávaly na těchto štítcích a zpracování probíhalo na elektromechanických strojích.

Největším krokem vpřed nejen pro databáze byl vývoj počítačů a informační techniky obecně.

Jelikož publikací pojednávajících o databázích vzniklo nepřeborné množství, nebude tato problematika detailně probrána. Účelem této kapitoly je uvedení čtenáře do problematiky, které povede k lepšímu pochopení databází jako prostředku pro uchování dat při dolování dat z databází.

3.1 Databáze – databázový systém

Databázový systém je systém, který slouží k efektivnímu ukládání, modifikaci a výběru velkého množství dat. Skládá se ze dvou částí: Systém řízení báze dat (SŘBD) a databáze samotné (báze dat – DB).

Systém řízení báze dat (SŘDB)

SŘBD je programové vybavení, které řídí přístup k datům v bázi dat. Umožňuje vytvářet databáze (báze dat), upravovat je a dotazovat se jich. Poskytovat přístup uživatelům a spravovat jejich přístup k vytvořeným databázím (bázím dat).

Báze dat (DB)

Báze dat je množina dat, která jsou vzájemně propojena. Data se do DB ukládají dle databázových modelů, které jsou popsány v kapitole 3.2.

Informace

Tento pojem může být chápán více směry. Její základní definici dle normy ISO [11] je možné zjednodušit. „Informace je sdělení přenášené médiem, které může mít podobu hmotnou i nehmotnou“. Informace se skládá z dat, jejich významu a struktury. Lze také říci, že jde o sdělitelný poznatek, který má smysl a snižuje nejistotu.

Data

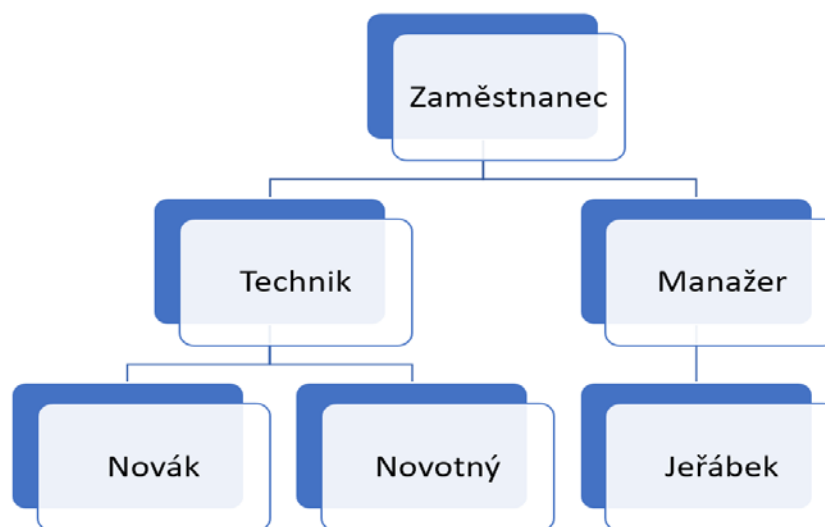
Data je možné definovat jako reprezentaci informace, kterou je možné přenášet a zpracovávat. Data mohou být reprezentována například v textové či číselné podobě.

3.2 Databázové modely dat

Z pohledu způsobu ukládání dat a vazeb mezi těmito daty se databáze dělí do několika typů, jež jsou uvedeny níže.

1. Hierarchický model dat

V tomto modelu jsou data (záznamy) organizována do podoby stromu, jak je možno vidět na obr. 2.1.



Obr. 2.1: Ukázka hierarchického modelu dat

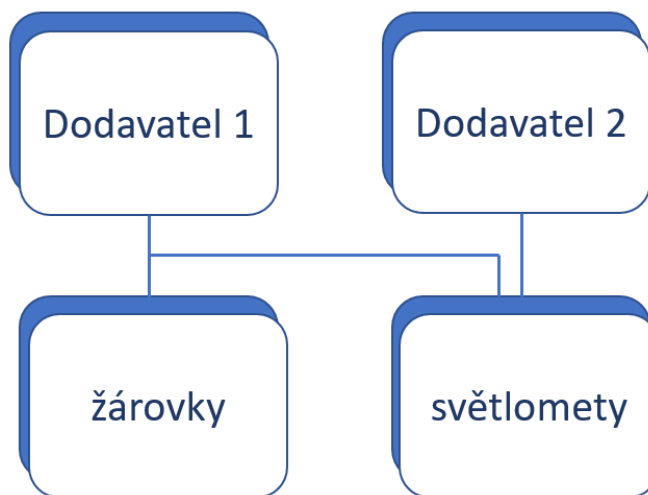
Každý záznam v tomto modelu představuje uzel stromové struktury. Vždy je jeden ze záznamů tzv. kořenem tohoto stromu (v případě obr. 2.1 jde o záznam zaměstnanec). Vztah mezi jednotlivými uzly je rodič – potomek. Z tohoto vztahu vyplývá omezení hierarchického modelu dat, a to možnost realizovat vztah pouze jeden k mnoha. Jeden uzel zde může mít pouze jeden rodičovské uzel.

Tento model byl využíván v době ukládání dat na magnetické pásky. V dnešní době se již příliš nepoužívá.

Nejznámější hierarchický model vytvořila společnost IBM pod názvem IMS (Information Management System) [12].

2. Síťový model dat

Síťový model organizuje data do tzv. sítě, čímž eliminuje omezení hierarchického modelu dat. V tomto modelu, na rozdíl od hierarchického lze realizovat mnohonásobné vztahy. Příklad síťového modelu dat vidíme níže na obr. 2.2.



Obr. 2.2: Ukázka síťového modelu dat

Nejznámějším využitím síťového modelu je IDMS (Integrated Database Management System) [13], který vytvořila společnost CA Technologies.

3. Relační model dat

Relační model sdružuje data do tzv. relací, tedy tabulek, které obsahují řádky. Jednotlivé řádky pro nás představují záznamy v tabulce, sloupce potom atributy. Všechny databázové operace jsou prováděny v těchto tabulkách.

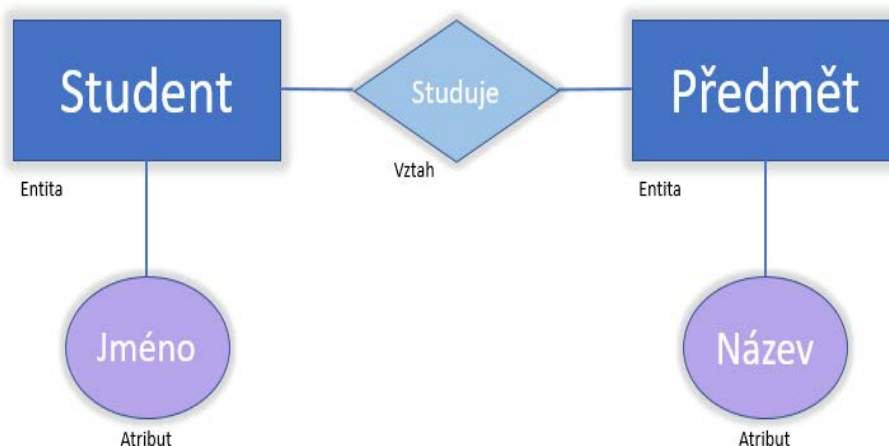
První implementací relačního modelu byl projekt System R laboratoří IBM [14].

O relační databázi blíže pojednává kapitola 3.3.

4. Sémantický model dat

Tento datový model přistupuje k datům odlišně než předchozí modely. Data jsou zde reprezentována objekty, množinami objektů a vztahů mezi objekty. Základními pojmy zde jsou entita (objekt), vztah, atribut (vlastnost).

Nejznámějším sémantickým modelem je Entity Relationship model (E-R model). Modelujeme zde například firmu (tedy pro nás důležitý objekt zájmu) jako relaci entit a vztahů mezi nimi. Entita znázorňuje jednoznačně identifikovatelnou věc existující samostatně, například objekt. Entity se označují podstatným jménem. Mezi jednotlivými entitami existují vztahy, které zachycují, jak jsou jednotlivé entity vztaženy k sobě navzájem. Tyto vztahy se označují slovesem. Entity i vztahy mohou obsahovat atributy (vlastnosti). Ukázku jednoduchého E-R diagramu je možno vidět na obr. 2.3.



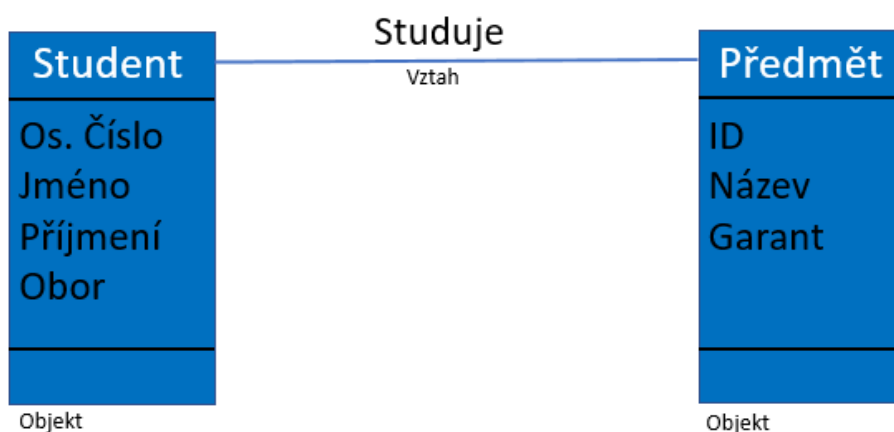
Obr. 2.3: Ukázka Entity Relationship modelu

5. Objektový model dat

U většiny klasických modelů databází jsou data zobrazována do předpřipravených struktur. V tomto modelu si nejprve připravíme objekt, který se podobá datům. Objekt zde představuje entitu. Model obsahuje také vztahy a metody. Metody určují, jaké činnosti je možné provádět s objektem. Vztahy určují, jak jsou jednotlivé objekty vztaženy k sobě navzájem. Příklad objektového modelu dat vidíme na obr. 2.4.

Objektového modelu využívají objektové databáze, které kombinují objektově orientované programování s databází. Poskytují možnosti programování databáze.

Tento model je vhodný pro data se složitou strukturou. Využívá se v geografických informačních systémech (GID) a počítačovém návrhu (CAD).



Obr. 2.3: Ukázka objektového datového modelu

6. Objektově-relační model dat

V tomto datovém modelu je relační model rozšířen o objektově orientovanou strukturu, tedy o objekty a konstruktory pro manipulaci s novými datovými typy. Data tedy modelujeme pomocí objektů, ale je zde zachována relační struktura. Místo záznamů (relační model) pracujeme s instancemi objektů (objektový model).

V objektově relačním modelu je zachována kompatibilita s relačními jazyky.

Podporu tohoto modelu dat nalezneme v databázích DB2, Oracle a Microsoft SQL Server.

3.3 Relační databáze


Dnes velice rozšířeným typem databáze je databáze relační. Tento typ databáze je založen na relačním modelu. Jeho hlavní strukturu tedy tvoří data uložená ve vzájemně propojených tabulkách. Tyto tabulky je možné také nazvat entity. Každá tabulka má sloupce a řádky. Sloupce vyjadřují vlastnosti (atributy) tabulky a musejí mít jedinečný název.

3.3.1 Tabulky relační databáze

Každá tabulka obsahuje sloupce a řádky. Sloupce vyjadřují vlastnosti (atributy) tabulky, musejí mít jedinečný název a datový typ. Řádky představují samotná data v tabulce.

Příklad:

Mějme tabulku s názvem Student. Tato tabulka bude obsahovat sloupce (vlastnosti): Os. Číslo (int), Jméno (nvarchar(20)), Příjmení (nvarchar(30)). V závorce za názvy sloupců je uveden jejich typ.

	Column Name	Data Type	Allow Nulls
	[Os.číslo]	int	<input type="checkbox"/>
	Jméno	nvarchar(20)	<input type="checkbox"/>
	Příjmení	nvarchar(30)	<input type="checkbox"/>
			<input type="checkbox"/>

Obr. 2.4: Návrh tabulky relační databáze

	Os.číslo	Jméno	Příjmení
1	1234	Jan	Novák
2	1235	Anna	Nováková
3	1236	Radim	Čákl

Obr. 2.5: Tabulka relační databáze s daty

Tabulku se všemi potřebnými náležitostmi vidíme níže na obr. 2.4. Nyní tabulky naplníme daty. Výsledná tabulka je zobrazena níže na obr. 2.5.

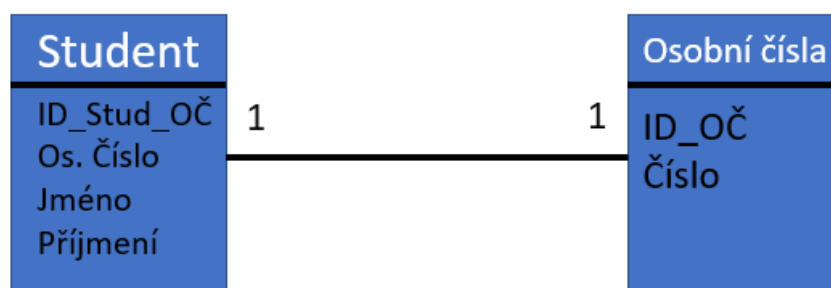
Aby bylo možné každý záznam (řádek) v tabulce jednoznačně rozpoznat, používáme tzv. primární klíč. Primární klíč je jednoznačný identifikátor daného záznamu. Tato hodnota musí být v rámci jedné tabulky jedinečná.

Vztahy mezi entitami (tabulkami)

Mezi jednotlivými tabulkami můžeme definovat vztahy (relace). Pro přehlednost mějme tabulku jedna a tabulku dvě.

1. Vztah jedna k jedné (1:1, one-to-one)

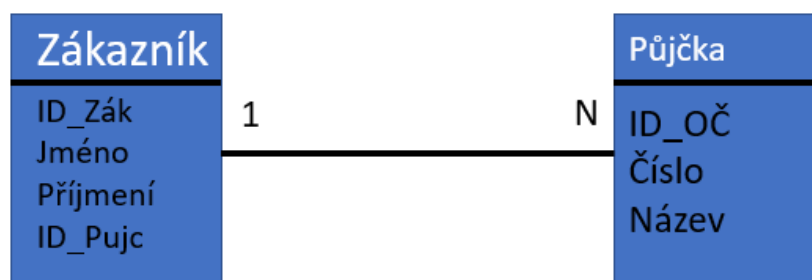
V tomto vztahu odpovídá jeden záznam v tabulce jedna pouze jednomu záznamu v tabulce dvě. Tuto relaci zajistíme pomocí unikátních klíčů v obou tabulkách. Příklad tohoto vztahu vidíme na obr. 2.6.



Obr. 2.6: Ukázka vztahu 1:1

2. Vztah jeden k mnoha (1:N, one-to-many)

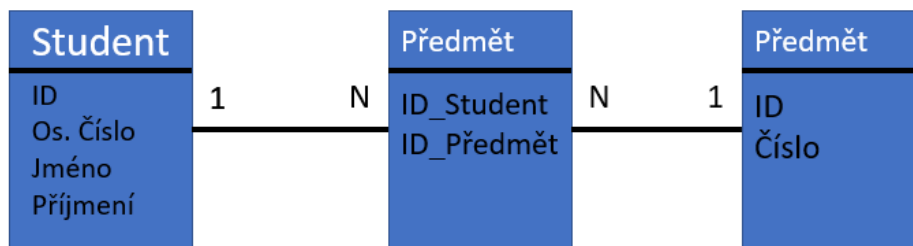
Zde odpovídá jeden záznam v tabulce jedna jednomu či více záznamům v tabulce dvě. Příklad vidíme na obr. 2.7.



Obr. 2.7: Ukázka vztahu 1:N

3. Vztah mnoho k mnoha (N:M, many-to-many)

U tohoto vztahu může odpovídat více záznamů v tabulce jedna více záznamům v tabulce dvě. V praxi se pro tento vztah používá dekompozice, což znamená, že je implementován pomocí pomocné tabulky. Vztah N:M rozložíme tedy na dva vztahy N:1. Ukázku implementace tohoto vztahu vidíme na obr. 2.8.



Obr. 2.8: Ukázka vztahu N:M

3.4 Dotazovací jazyky

Pro usnadnění práce s databází slouží uživateli dotazovací jazyky. Jejich úkolem je pomoci uživateli s vyhledáním, vložením či editací požadovaných dat.

Dotazovací jazyky jsou děleny na procedurální a neprocedurální. Procedurální jazyky jsou složitější a vyžadují zpravidla zadání správného algoritmu pro získání požadovaného výsledku. Neprocedurální jazyky jsou naopak mnohem jednodušší, jelikož vyžadují pouze zadání podmínek, které má požadovaný výsledek splňovat.

V praxi je možné setkat se především s jazyky SQL (Structured Query Language) a QBE (Query By Example), jejichž vysvětlení je uvedeno níže.

Jazyk SQL (Structured Query Language – strukturovaný dotazovací jazyk)

Jde o strukturovaný jazyk vhodný pro uživatele databáze. Jeho struktura není příliš náročná.

Příkazy jazyka SQL je možné rozdělit do čtyř skupin viz níže.

1. DDL (Data Definition Language).

Příkazy v této skupině je možné vytvářet či upravovat strukturu databáze.

Příklad: CREATE, DROP

2. DML (Data Manipulation Language).

Příkazy v této skupině složí k vyhledávání, ukládání a mazání dat v databázi.

Příklad: SELECT, INSERT, UPDATE, DELETE

3. DCL (Data Control Language).

Jedná se o příkazy ke správě uživatelských práv a rolí.

Příklad: REVOKE

4. TCL (Transaction Control Language).

Těmito příkazy je možné spravovat databázové transakce.

Příklad: BEGIN, ROLLBACK

Jazyk QBE (Query By Example)

QBE je prvním grafickým dotazovacím jazykem. Dotaz v tomto jazyce se vytváří pomocí znaků vepisovaných do databázového formuláře. Tento jazyk se dnes již používá zřídka.

3.5 Multidimenzionální databáze

Jelikož výsledná aplikace pracuje s multidimenzionálními daty, je nutné seznámit se s databází pro jejich ukládání.

V předchozí kapitole byla stručně vysvětlena problematika relační databáze. Tento typ databáze je v oblasti multidimenzionálních dat využit jako zdroj dat, jež budou dále zpracována a uložena v multidimenzionální databázi.

V multidimenzionální databázi jsou data organizována do multidimenzionálních struktur. Z tohoto typu databáze snáze získáme sumarizované a agregované údaje, tedy informace. Používáme zde nenormalizované tabulky dvou typů: tabulky faktů a tabulky dimenzí. Do multidimenzionální databáze se ukládají již předzpracovaná data.

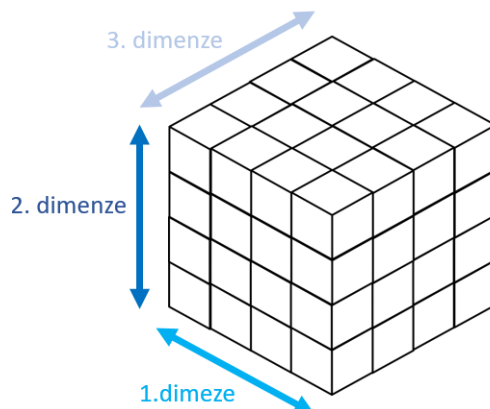
Tyto analytické databáze je možné označit i pojmem OLAP (Online Analytical Processing). Tento pojem označuje struktury údajů a analytické služby sloužící k analýze většího množství údajů.

3.5.1 Multidimenzionální databázový model

V relačních databázích pracujeme s tabulkou, která obsahuje řádky a sloupce. Jelikož zde pracujeme s více dimenzemi, je tento model nedostačující. Nejobvyklejší datovou strukturou multidimenzionálního databázového modelu je krychle.

Příklad multidimenzionálního datového modelu

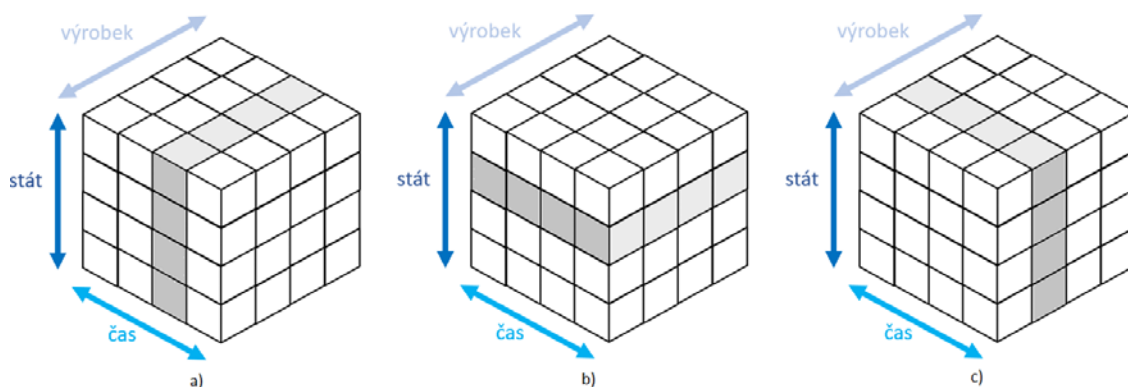
Ačkoli multidimenzionální databázový model může mít, na rozdíl od geometrické krychle, více dimenzí, ukážeme si příklad tohoto modelu právě na modelu krychle.



Obr. 2.9: Multidimenzionální krychle

Obecný příklad databázového modelu se třemi dimenzemi, též nazývaný jako multidimenzionální krychle (kostka) někdy též OLAP krychle nalezneme na obr. 2.9.

Údaje se v multidimenzionální krychli nacházejí v průnicích jednotlivých dimenzí. Pro příklad mějme tyto tři dimenze: 1. čas, 2. stát, 3. výrobek. Příklad takovéto databázové krychle vidíme na obr. 2.10. Údaje v databázi s těmito dimenzemi je možné analyzovat třemi směry, což vidíme na již zmíněném obr. 2.10. Jde o analýzu údajů za určité časové období (obr. 2.10a) například pro vyhodnocení návštěvnosti webových stránek, dle jednotlivých států (obr. 2.10b) například pro vyhodnocení úspěšnosti obchodu v jednotlivých státech a dle jednotlivých výrobků nebo skupiny výrobků (obr. 2.10c) například pro úspěšnost prodeje nabízených výrobků, počty prodaných kusů apod..



Obr. 2.10: Multidimenzionální krychle: a) Analýza údajů za časové období, b) Analýza údajů dle státu, c) Analýza údajů dle výrobku nebo skupiny výrobků

3.5.2 Dotazovací jazyk MDX (Multidimensional Expressions)

Jazyk MDX používaný v multidimenzionální databázi je ekvivalentem jazyka SQL, který je používán v relačních databázích. Pomocí dotazů MDX je možné vypsát údaje z multidimenzionální krychle do dvojrozměrné tabulky.

Struktura jazyka MDX

Podobně jako jazyk SQL i jazyk MDX používá základní příkaz SELECT. Jeho struktura je následující:

```
SELECT [* | (<specifikace osy> [, <specifikace osy> ...]]
FROM [<specifikace krychle>]
Případně je možné přidat klauzuli WHERE:
WHERE [<specifikace řezu (upřesňující pomyslná osa)>]
```

Programový kód 1: Struktura příkazu SELECT jazyka MDX

Po úspěšném vytvoření multidimenzionální databáze, jež je popsáno v následující kapitole, je možné testovat dotazy jazyka MDX v programu Microsoft SQL Server

Management Studio. Ukázky dotazů a použití jazyka MDX je možno nalézt v dokumentaci na stránkách firmy Microsoft [15].

3.5.3 Vytvoření OLAP krychle z relační databáze

Pro práci s velkým množstvím dat je výhodné použití multidimenzionální databáze, tedy vytvoření OLAP krychle. Abychom mohli tuto strukturu (OLAP krychli) vytvořit, musíme data, která chceme analyzovat tzv. očistit, určit dimenze a řadu dalších parametrů. K tomuto účelu potřebujeme databázový program s daty, který chceme analyzovat a program pro vytvoření samotné OLAP krychle. Níže je uveden souhrn všech potřebných nástrojů. Nutno poznamenat, že k vytvoření multidimenzionální databáze je potřebný analytický server, který je třeba nainstalovat spolu s MS SQL Serverem. Pokud u verze Express nebude tato možnost poskytnuta, je možné stáhnout verzi Developer. Tato verze MS SQL Serveru ovšem vyžaduje účet Microsoft.

V textu uvedeném níže je pro názornost vytvořena OLAP krychle pomocí programu Microsoft Visual Studio. Jde o ukázkou pro lepší pochopení konceptu OLAP krychle. Pro výslednou aplikaci je tato krychle vytvořena pomocí kódu, jež je popsán v kapitole 7.

Softwarové požadavky pro tvorbu OLAP krychle

Microsoft Visual Studio – Verzi Community je možné zdarma získat zde [16]

Microsoft SQL server – Verzi Express je možné zdarma získat zde [17]

Microsoft SQL server – Verzi Developer je možné získat zde [18]

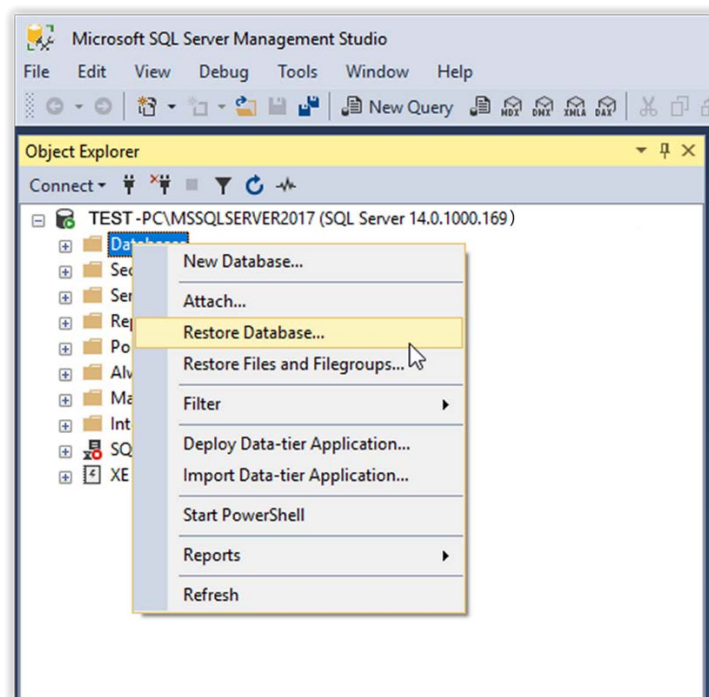
Doplňek SQL Server data tools (SSDT) – ke stažení zde [19]

Testovací databáze Adventure works DW 2016 – ke stažení zde [20]

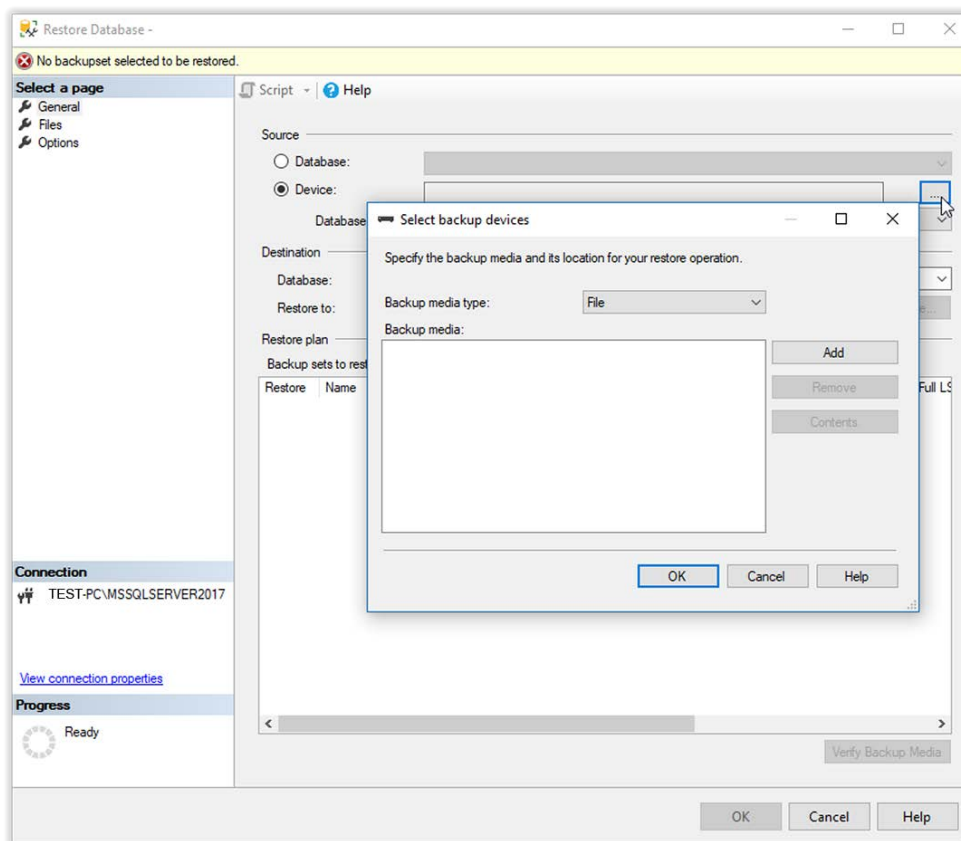
Nahrání databáze Adventure works DW2016 do MS SQL Serveru

Tuto databázi lze stáhnout (viz výše), jako soubor AdventureWorksDW2016.bak. Jde o soubor zálohy databáze, proto následujícím postupem bude její obnovení v MS SQL Serveru.

Spustíme program Microsoft SQL Server Management Studio. Přihlásíme se dle údajů zadaných při instalaci. V levé části se nachází menu databázového serveru. Klikneme pravým tlačítkem na složku *Database* a zvolíme *Restore Database* dle obr. 2.11, jež je uveden níže.



Obr. 2.11: Obnovení databáze



Obr. 2.12: Výběr souboru zálohy databáze

V nově otevřeném okně zvolíme zaškrtnutí políčko Device a na téže řádce najdeme tlačítko se třemi tečkami na něž klikneme. V následujícím okně klikneme

na tlačítko Add vpravo (obr. 2.12), vyhledáme stažený soubor AdventureWorksDW2016.bak a potvrdíme OK.

Do okna se seznamem databází určených k obnovení bude načtena záloha zvolené databáze. Toto okno potvrdíme tlačítkem OK. Po úspěšném proběhnutí klikneme pravým tlačítkem na složku Databases a zvolíme Refresh. Nyní, pokud klikneme na symbol plus vlevo vedle složky Databases uvidíme v zobrazeném seznamu databází databázi AdventureWorksDW2016.

Vytvoření krychle OLAP

Pro vytvoření multidimenzionální kostky použijeme program Visual Studio s doplňkem SSDT, kde vytvoříme šablonu kostky (vytvoření kostky programově). Tuto kostku poté pomocí programu Microsoft Visual Studio “promítneme” do Analytického serveru. Z OLAP kostky v programu vytvoříme OLAP kostku v databázi.

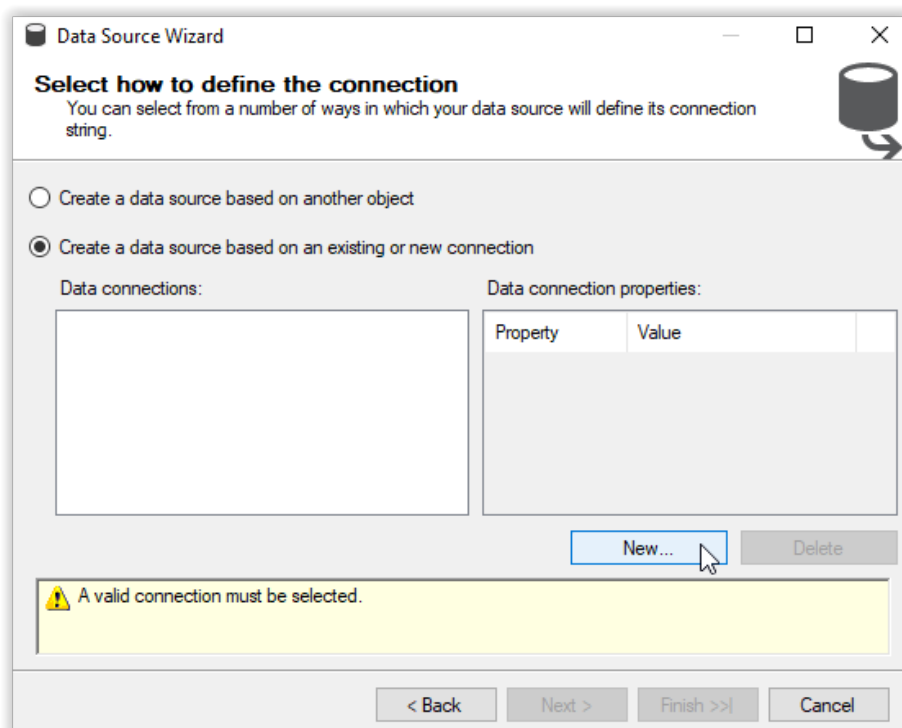
Začneme s vytvořením nového projektu. Spustíme program Visual Studio, klikneme vpravo nahoře na položku Soubor a vybereme *Nový* a dále *Projekt*. Po nainstalování doplňku SSDT přibude v levém menu pod rozevíracím seznamem Nainstalované, položka *Business Intelligence*. Po dvojím kliknutí na tuto položku se zobrazí dvě podúrovně *Analysis Services* a *Reporting Services*. Dále klikneme na *Analysis Services*, jelikož budeme data analyzovat. Uprostřed zobrazeného okna se objeví seznam projektů. Vybereme projekt *Analysis Services Multidimensional and Data Mining Project*. Název pro projekt je možné vybrat jakýkoli. V tomto ukázkovém projektu je projekt nazván OLAPTest. Veškeré položky v tomto projektu jsou v anglickém jazyce i databáze použitá pro tento účel je v anglickém jazyce. Je tomu tak proto, že česká verze doplňku SSDT není k dispozici a ani rozsáhlá databáze potřebná pro tyto účely není k dispozici. V místech potřebných pro pochopení jednotlivých kroků bude uveden významový překlad a vysvětlení pojmů.

V okně Průzkumník řešení nalezneme předpřipravené složky. Pokud se průzkumník nezobrazuje, je možné jej zobrazit kombinací kláves ctrl, alt, L nebo jej najdeme v menu Zobrazit, Průzkumník řešení. Postupně budeme vytvářet data ve složkách *Data Sources*, *Data Sources Views*, *Cubes* a *Dimensions*. Postup, pokud nebude uvedeno jinak, sestává vždy z kliknutí pravým tlačítkem na požadovanou složku a zvolením *New* za nímž následuje název složky například: *New Data Source*. Dále bude popsáno, jak postupovat pro vytvoření potřebné položky v nově otevřeném okně. Všechny kroky v jednotlivých složkách uvedené níže, je třeba provádět postupně, tak jak jsou popsány, jelikož na sebe plynule navazují a jsou na sobě závislé.

1. Data Source (zdroj dat)

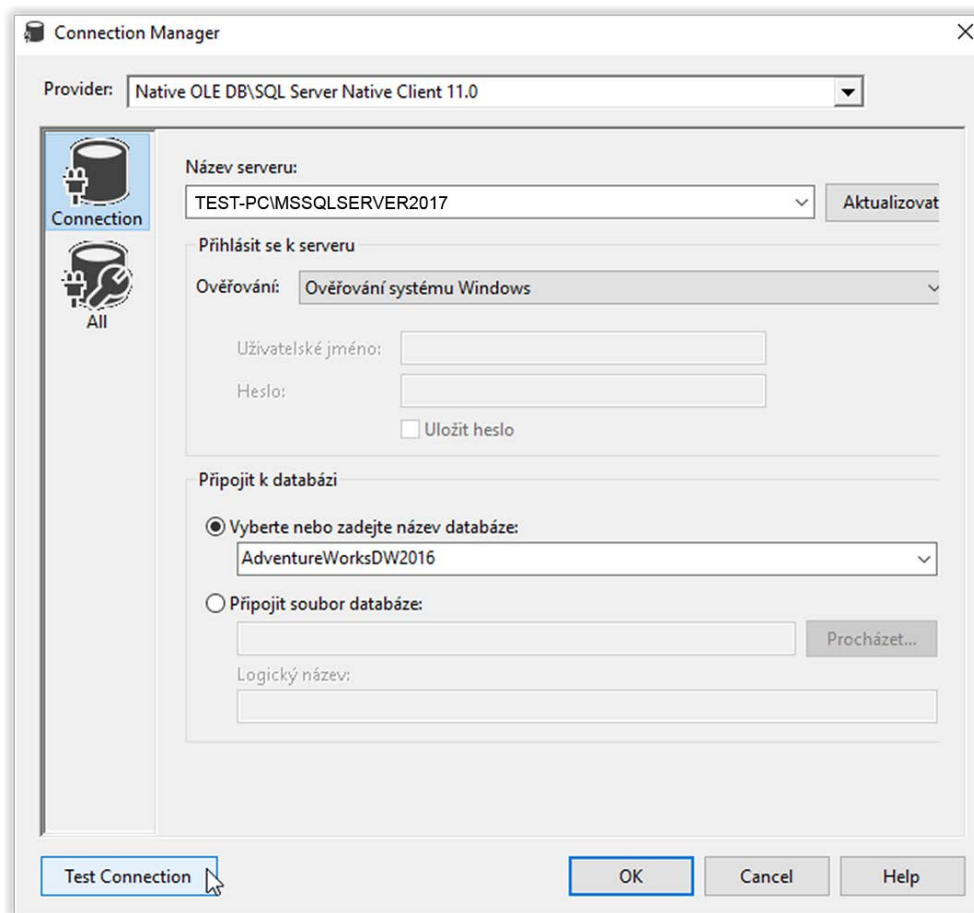
V prvním kroku vytvoříme zdroj dat pro vytvoření multidimenzionální krychle.

Po zobrazení okna s uvítací obrazovkou pomocníka pro vytvoření nového zdroje dat přeskočíme uvítací obrazovku a kliknutím na tlačítko *Next* se posuneme k prvnímu kroku. Zde zvolíme pole *Create a data source based on an existing or new connection* a vpravo dole klikneme na tlačítko *New* (obr. 2.13). Jde o přidání nového připojení ke zdroji dat, k databázi. Tento krok předpokládá běžící databázový server.



Obr. 2.13: Připojování zdroje dat

V novém okně zadáme název serveru a níže název databáze, jak je uvedeno na obr. 2.14 viz níže. Spojení s databází je možné otestovat tlačítkem *Test connection* vlevo dole. Pokud se po jeho stisknutí objeví malé okno s textem *Test connection succeeded* je spojení v pořádku a je možné vše potvrdit stisknutím tlačítka *OK* (okno *Test connection*) a *OK* (okno *Connection manager*).



Obr. 2.14: Připojení databáze jako zdroje dat

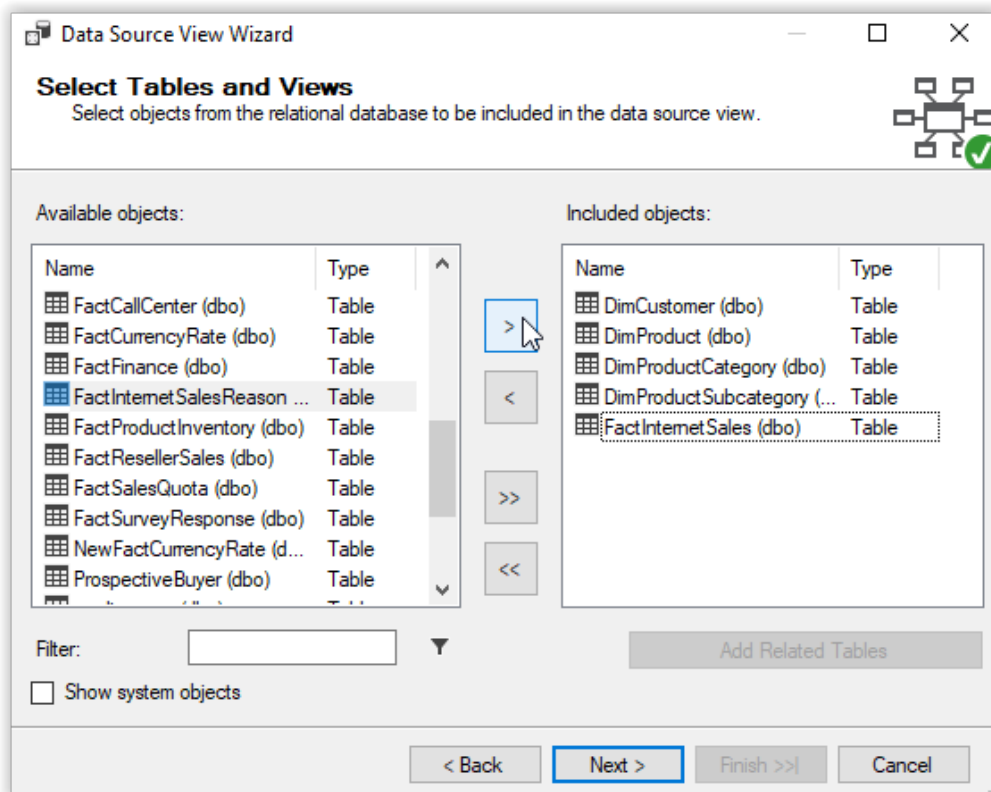
Po vytvoření datového zdroje se tento zdroj objeví v poli *Data Connections*. Pokračujeme tlačítkem *Next*. Následuje volba přihlášení ke zdroji dat. Zvolíme *Use the service account* a klikneme *Next*. Zbývá už jen pojmenovat nově vytvořený zdroj dat. V poli pro název vidíme automaticky předvyplněný název totožný s názvem databáze. Potvrdíme tlačítkem *Finish*.

2. Data Sources Views (zdroj dat)

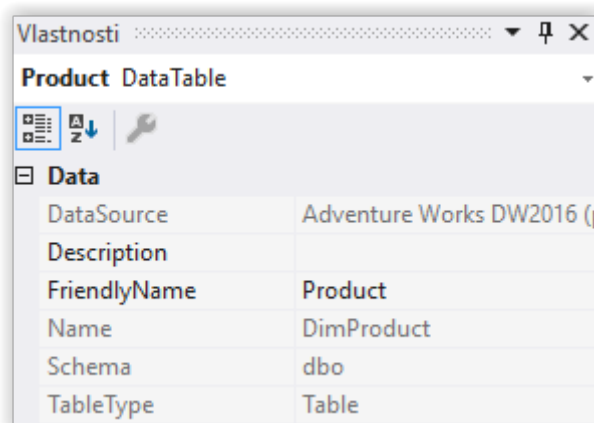
Zde vytvoříme tzv. pohled. Jde o grafické znázornění připojených tabulek ze zdrojové databáze.

V nově otevřeném okně opět přeskočíme uvítací obrazovku stiskem *Next*. Následně je v poli *Relational data source* zobrazen zdroj dat vytvořený v předchozím kroku. Pokud bychom měli více zdrojů dat, je možné zvolit jiný datový zdroj kliknutím na tlačítko *New Data Source*. V tomto případě nic měnit nebudeme a přesuneme se k dalšímu kroku stiskem *Next*. V novém zobrazení vidíme vlevo názvy tabulek v připojené databázi (pole *Available objects*) a vpravo zatím prázdné místo určené pro vybrané tabulky (pole *Included objects*). Vybereme a poté pomocí tlačítka se symbolem > přesuneme požadované tabulky z levé části do pravé dle obr. 2.15. Po výběru správných tabulek se přesuneme k dalšímu kroku kliknutím

na tlačítko *Next*. Následujícím a již posledním krokem je pojmenovat vytvořený datový pohled. V poli *Name*, což je pole pro název pohledu, máme již předvyplněn stejný název, jako název databáze. Potvrdíme stiskem tlačítka *Finish*.



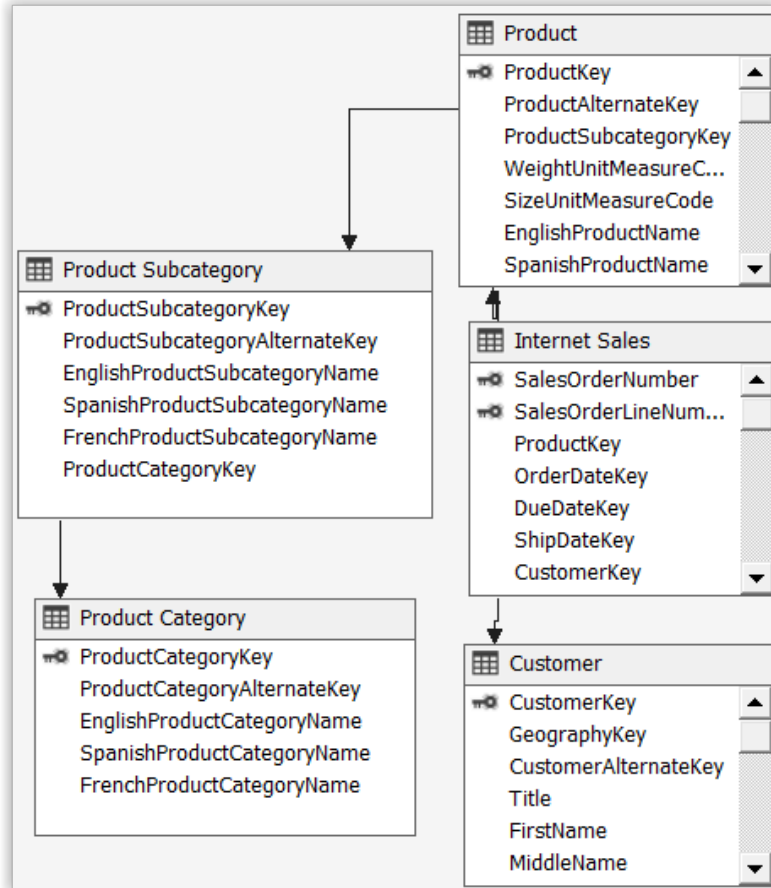
Obr. 2.15: Výběr tabulek do pohledu (Data Source View)



Obr. 2.16: Vlastnosti tabulky v pohledu (Data Source View)

Po vytvoření datového pohledu je tento rovnou otevřen a je možné shlédnout návaznosti vložených tabulek a všechny položky, které obsahují. Tento datový pohled slouží k upravení názvu tabulek. Jelikož všechny tabulky začínají slovem Dim nebo Fact, není orientace v nich příliš dobrá. Odstraníme tedy tato slova z názvů tabulek. Docílíme toho tak, že klikneme na tabulku pravým tlačítkem myši a vybereme možnost *Vlastnosti*. Zobrazí

se okno s vlastnostmi dané tabulky. Ve vlastnostech tabulky nalezneme položku *Friendly name* (přátelské jméno – název tabulky přijatelný pro běžného uživatele), a jméno tabulky změníme (obr. 2.16). Takto přejmenujeme názvy všech tabulek, jak je uvedeno na obr. 2.17.



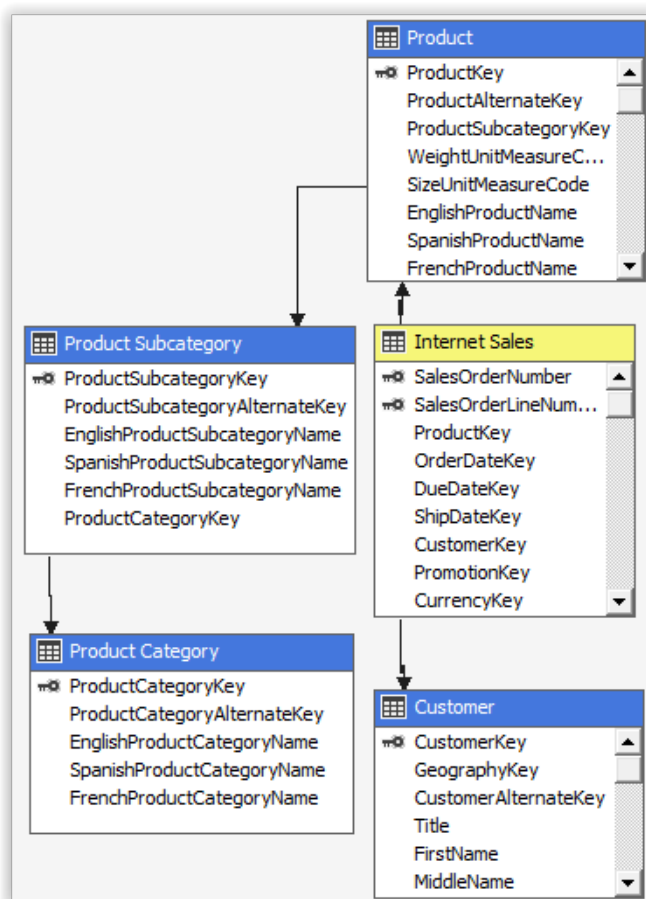
Obr. 2.17: Přejmenování tabulek v pohledu (Data Source View)

3. Cubes (krychle(OLAP))

Ve složce s názvem *Cubes* (krychle) bude vytvořena již samotná krychle OLAP. Všemi dosud uvedenými kroky jsme připravili potřebná data a nyní je přetvoříme do multidimenzionální krychle.

Po odkliknutí uvítacího okna pomocníka pro vytvoření OLAP kostky volíme způsob jejího vytvoření. Zvolíme možnost *Use existing tables*. Znamená to, že pro vytvoření OLAP krychle použijeme tabulky, jež jsme vybrali v předchozím kroku. Svoji volbu potvrdíme tlačítkem *Next*. Následně vybíráme tzv. *Measure Group tables*. Jde o tabulku (skupiny tabulek), jež bude sloužit jako měrná jednotka. Zvolíme tabulku *Internet Sales*, která obsahuje například hodnotu *Unit price* nebo *ProductStandardCost*, což jsou ceny produktů. Po potvrzení tlačítkem *Next* vybereme, které hodnoty ve vybrané tabulce budou měrnými. Mohli bychom vybrat všechny hodnoty z tabulky, ale bylo by to zbytečné a nepřehledné. V tomto případě vybereme hodnoty *Order Quantity*, *Unit Price*, *Sales Amount* a *Internet Sales Amount*.

Stiskem tlačítka *Next* přejdeme k dalšímu kroku. Zde jsou zobrazeny průvodcem předpřipravené dimenze budoucí multidimenzionální kostky. Vše potvrdíme stiskem *Next*. Následuje shrnutí nastavení z předchozích kroků. Pokud je vše v pořádku, kliknutím na tlačítko *Finish* vytvoříme OLAP kostku.



Obr. 2.18: Pohled strukturu vytvořené krychle OLAP (Cubes)

Po vytvoření krychle OLAP je otevřen pohled na strukturu nové krychle. Uprostřed lze vidět zvolenou měrnou tabulku a okolo ní ostatní vybrané tabulky. Všechny uvedené tabulky jsou vzájemně propojeny dle vztahů mezi nimi. Možný výsledek lze vidět na obr. 2.18. Ve složce *Dimensions* lze vidět nově vytvořené dimenze.

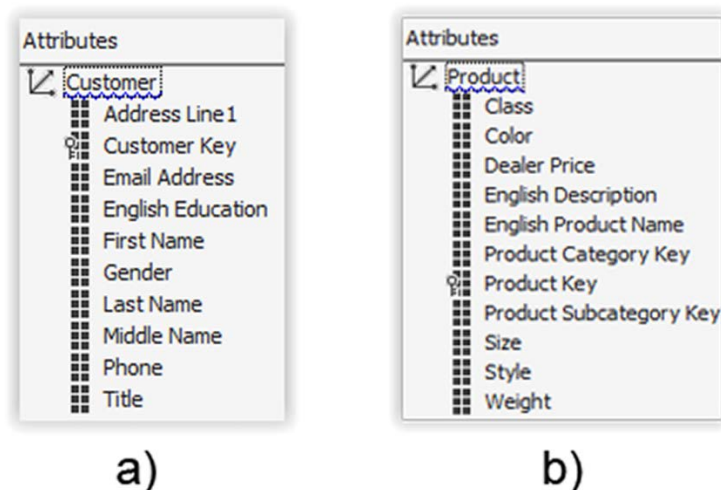
4. Dimensions (dimenze multidimenzionální krychle)

Tato složka slouží k vytvoření dimenzí multidimenzionální krychle a je před vytvořením krychle OLAP prázdná. Při vytváření dimenzí existují dvě možnosti. První možností je vytvořit dimenze ručně, každou zvlášť. Druhou, a jednodušší variantou, je nechat dimenze vytvořit pomocníkem pro tvorbu multidimenzionální krychle. Novou dimenzi je možné vytvořit, pokud klikneme pravým tlačítkem na složku s názvem *Dimensions* a vybereme *New*

Dimension. Jako u vytváření předchozích položek ve složkách i zde je pomocník, který usnadní vytvoření jednotlivých dimenzí. V případě tohoto příkladu necháme vytvořit dimenze automaticky. Pokud budeme postupovat dle návodu uvedeného výše, jsou již dimenze vytvořeny.

Dimenze již sice jsou vytvořeny, ale jen se základními položkami, které vybral průvodce (pomocník) při tvorbě kostky. Nyní je lze doplnit.

Jednotlivé dimenze je možné upravovat po dvojím kliknutí na jejich název. Základní okno pro práci s danou dimenzí je rozděleno do tří částí: *Attributes*, *Hierarchies* a *Data Source View*. V části *Attributes* vidíme název dimenze a pod ním její atributy (vlastnosti). Tyto vlastnosti doplníme ze třetí části okna, *Data Source View*. Zde se nachází tabulka s většinou stejným názvem jako má dimenze. Z této tabulky je možné přenést atributy do první části, tedy k atributům dimenze. Data přeneseme způsobem drag and drop (chyt' a pusť). Pro tento příklad přeneseme: dimenze *Customer* obr. 2.19a, dimenze *Product* obr. 2.19b. V dimenzi *Internet Sales* necháme vše tak, jak je.



Obr. 2.19: Úprava vlastností dimenzí: a) dimenze Customers, b) dimenze Products

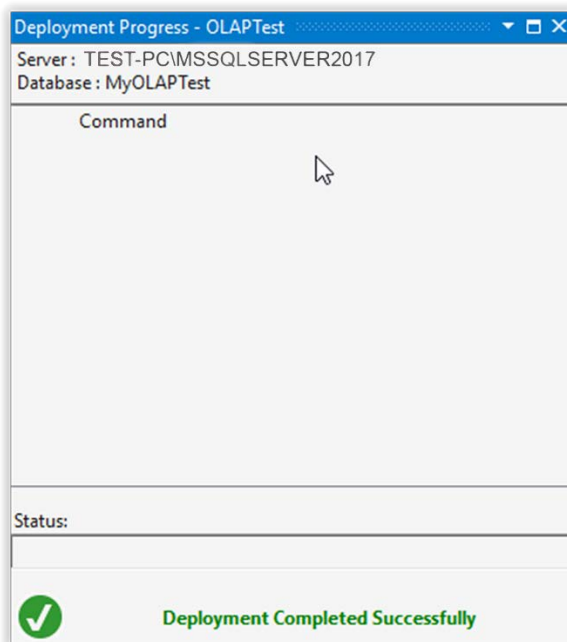
Nyní přistoupíme k převedení OLAP kostky v programu Visual Studio do databáze. K tomuto kroku potřebujeme drobný zásah do Microsoft SQL Serveru. Musíme zde vytvořit uživatele pro přístup k analytickému serveru. K vytvoření uživatele je nutné znát jméno uživatele, na kterého se přihlašuje služba analytického serveru. Název tohoto uživatele nalezneme u jeho služby. Ve vyhledávání operačního systému Microsoft Windows napíšeme slovo *Služby* (případně *Services*) a spustíme. Zde vyhledáme službu s názvem *SQL Server Analysis Service* (<Název serveru>). Vpravo vedle názvu služby se nachází položka *Účet* pro přihlášení, ve které nalezneme požadované jméno účtu.

Se znalostí názvu účtu, pod kterým je spuštěna služba analytického serveru, je možné vytvořit uživatele. Spustíme tedy Microsoft SQL Server Management

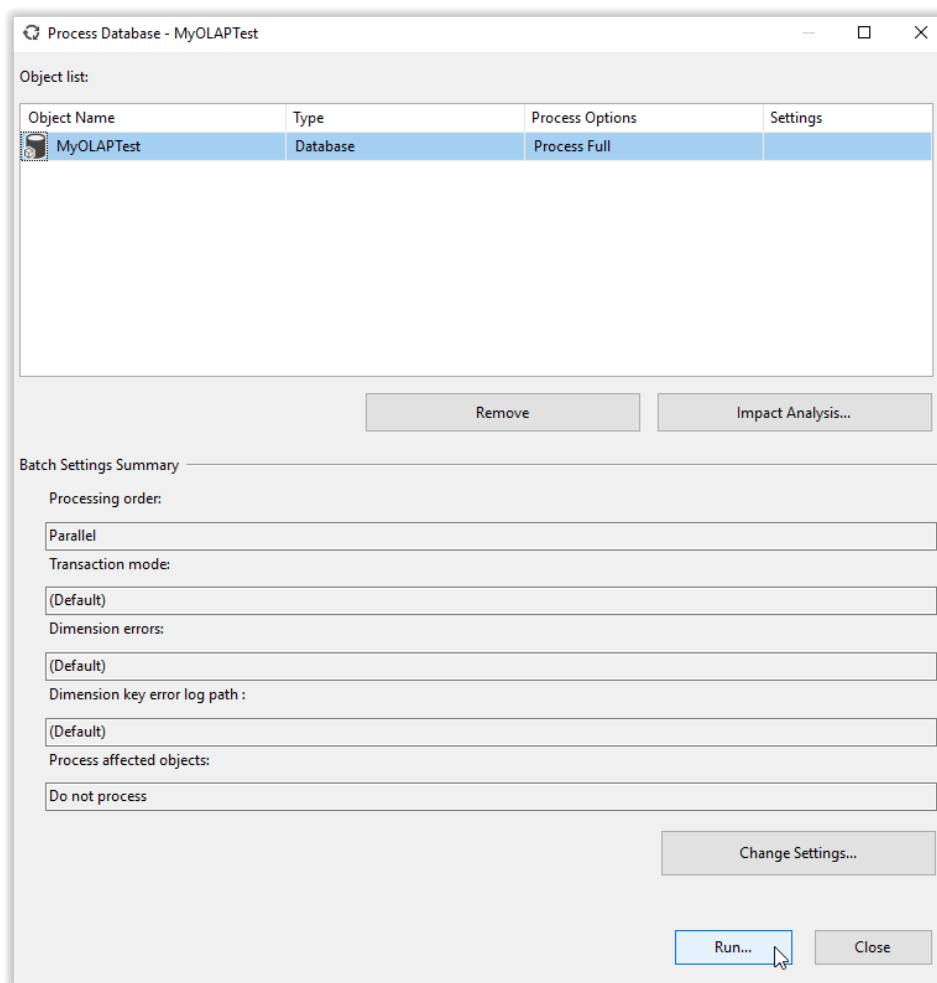
Studio a přihlásíme se k databázovému serveru. Ve složce *Security* vybereme složku *Logins*, klikneme na ni pravým tlačítkem a zvolíme *New Login*. V nově otevřeném okně vyplníme jméno uživatele stejné, jako jméno uživatele pro výše zmíněnou službu. V pravém seznamu vybereme položku *User Mapping* a zaškrtneme políčko vlevo vedle názvu databáze AdventureWorksDW2016. Následně vytvoříme definovaného uživatele potvrzením tlačítkem OK.

Po provedení výše uvedených kroků je nutné nastavit parametry pro vytvoření krychle OLAP v databázi. V programu Microsoft Visual Studio klikneme v *Průzkumníku řešení* pravým tlačítkem na název projektu a vybereme *Vlastnosti*. V levém seznamu vybereme záložku *Deployment*. Zde musíme upravit položku *Server*. Výchozí hodnota této položky je localhost. Místo ní vložíme celý název serveru, ke kterému se připojujeme. Můžeme také u položky *Database* změnit název budoucí databáze. Úpravy potvrdíme nejprve stiskem tlačítkem *použít* a poté *OK*. Znovu klikneme pravým tlačítkem na název projektu a zvolíme *Process...* Dotaz na nasazení (*deploy*) projektu potvrdíme. Dojde k vytvoření struktury kostky v databázi.

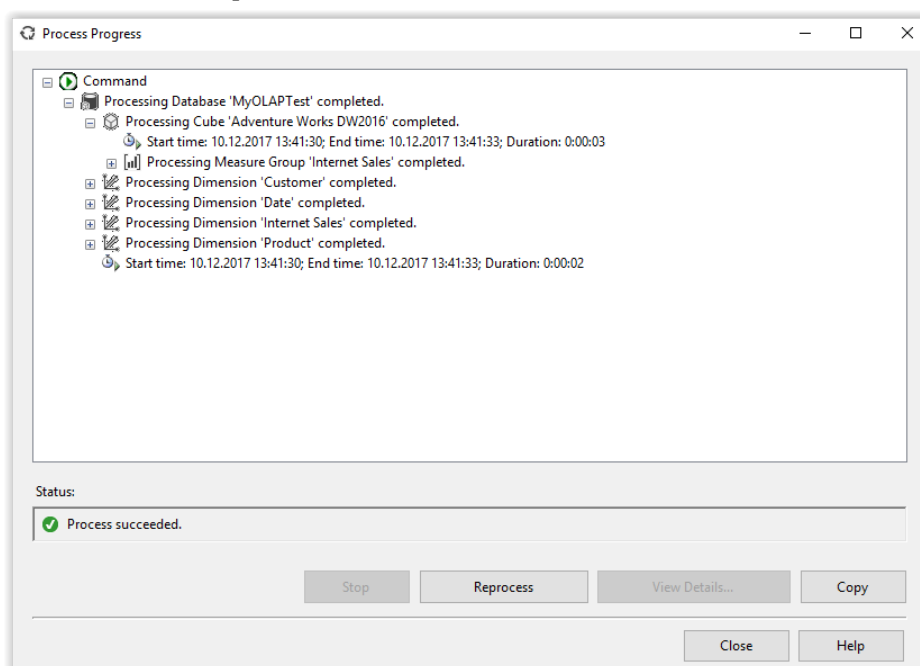
Pokud první krok proběhne v pořádku objeví se dvě okna: obr. 2.20 s potvrzením o nasazení a obr. 2.21 pro zpracování dat. Tato okna jsou zobrazena níže. Ve druhém zobrazeném okně klikneme na tlačítko Run. Po úspěšném zpracování bude nově zobrazené okno vypadat obdobně jako na obr. 2.22 viz níže. Všechna zobrazená okna můžeme zavřít. Vytvoření OLAP kostky, jako multidimenzionální databáze, je tímto dokončeno.



Obr. 2.20: Potvrzení nasazení OLAP kostky



Obr. 2.21: Zpracování dat do multidimenzionální databáze



Obr. 2.22: Úspěšné zpracování dat multidimenzionální databáze

4 ZOBRAZENÍ MULTIDIMENZIONÁLNÍCH DAT

Multidimenzionální data, především pokud je jich velký počet, mohou působit nekonzistentně a s jejich přibývajícím množstvím se pochopení informací v nich obsažených stává obtížné. Kvůli této skutečnosti se přistupuje k jejich analytickému zobrazení.

Data můžeme zobrazovat v podobě uspořádaných dat v tabulkách či čistě vizuální cestou pomocí grafů. Interpretace dat vždy záleží na účelu, ke kterému má sloužit. Je-li nutné znát konkrétní hodnoty, lze využít rozličně formátovaných tabulek. Jde-li naopak pouze o vizuální přehled využívá se grafu. Jelikož ve výsledné aplikaci budou data zobrazena graficky, zaměříme se na tento typ zobrazení.

4.1 Grafické zobrazení multidimenzionálních dat

Jak již název napovídá, u multidimenzionálních dat pracujeme s jednou či více dimenzemi. Popis multidimenzionálního modelu databáze nalezneme v kapitole 3.5. Každá taková databáze obsahuje již zmíněné dimenze a fakta. Dimenze určují jistý směr pohledu na data (například čas, komoditu apod.), a skládají se z několika atributů, jež popisují daný směr pohledu. Fakta jsou samostatnými údaji. Jde o průsečíky dat jednotlivých dimenzí nebo o tzv. agregace, což jsou výpočtem zjištěné hodnoty z dostupných dat. Jde například o součet, počet hodnot nebo průměr. Níže jsou uvedeny možnosti zobrazení těchto dat v grafu. Všechny uvedené typy grafu lze zobrazit ve dvojrozměrném i trojrozměrném provedení.

Pro zobrazení dat v grafu jsou nejčastěji využity tyto jeho typy:

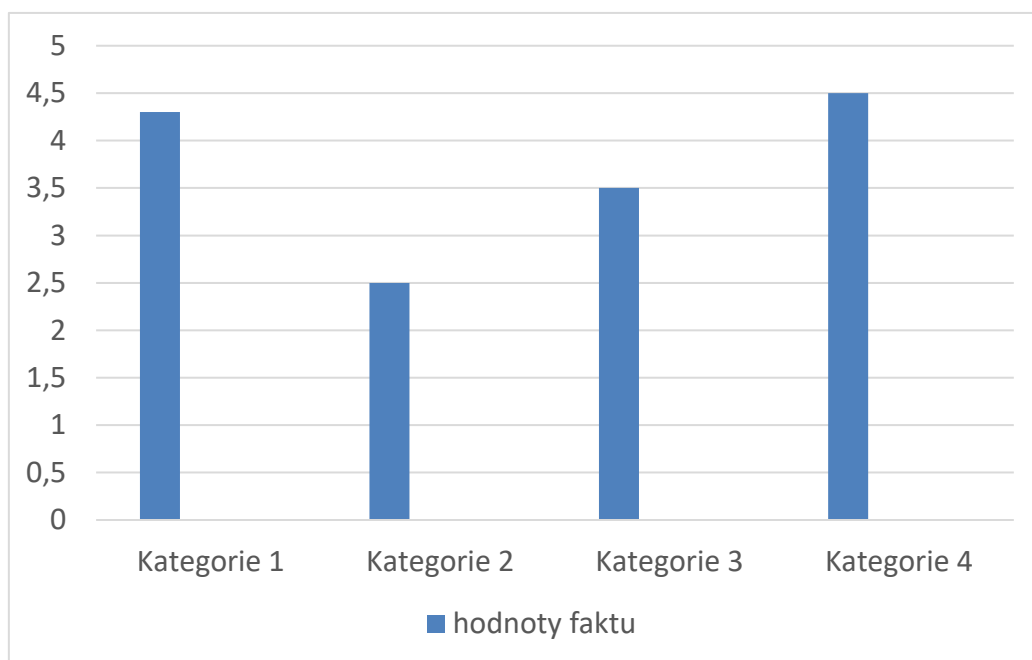
1. Sloupcový graf

Sloupcový graf se skládá ze sloupců, jež jsou většinou obdélníkového tvaru, ale mohou mít tvar kužele či trojúhelníku. Délky těchto sloupců vyjadřují množství veličiny, jež představují. Příklad sloupcového grafu můžeme vidět níže na obr. 3.1. Jde o graf, jenž obsahuje jednu dimenzi a jeden fakt. Dimenzi zde představuje Kategorie a faktem zde například může být počet zboží v dané kategorii.

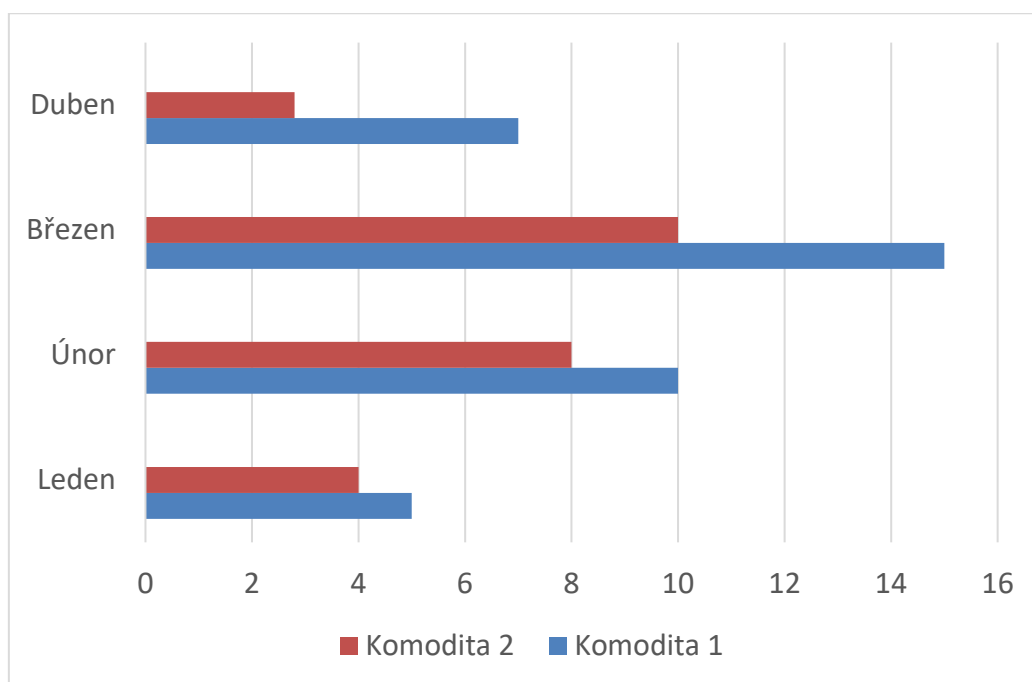
Často je také využíván tzv. pruhový graf, jež vychází ze sloupcového typu grafu, ale jeho sloupce jsou vedeny vodorovně. Příklad tohoto grafu je možné vidět níže na obr. 3.2. Jde o graf, jenž obsahuje dvě dimenze a jeden fakt. Jednu dimenzi představují měsíce v roce a druhou dimenzi komodity. Faktem může být například počet prodané komodity v daném měsíci v roce.

Hodnota faktu je znázorněna délkou sloupce (řádku). Množiny hodnot

využité pro tento příklad jsou Komodita 1 $K_1 = \{5, 10, 15, 7\}$ a Komodita 2 $K_2 = \{4, 8, 10, 3\}$.



Obr. 3.1: Sloupový graf zobrazující jednu dimenzi a jeden fakt

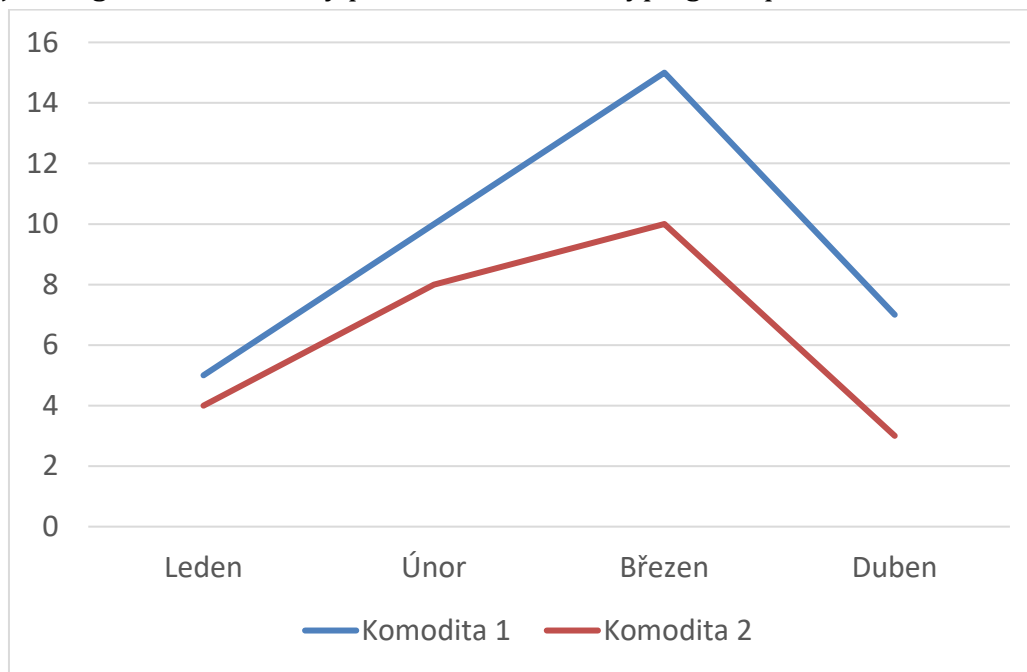


Obr. 3.2: Pruhoý (Sloupový) graf zobrazující dvě dimenze a jeden fakt

2. Spojnicový graf

Spojnicový graf zobrazuje data jako množinu hodnot spojených čarou. Využívá se především v případě, kdy je nutné zobrazit data v čase.

Příklad spojnicového grafu vidíme na obr. 3.3. Jde zde o stejná data jako u grafu 3.2. Lze tedy porovnat vhodnost typu grafu pro danou situaci.

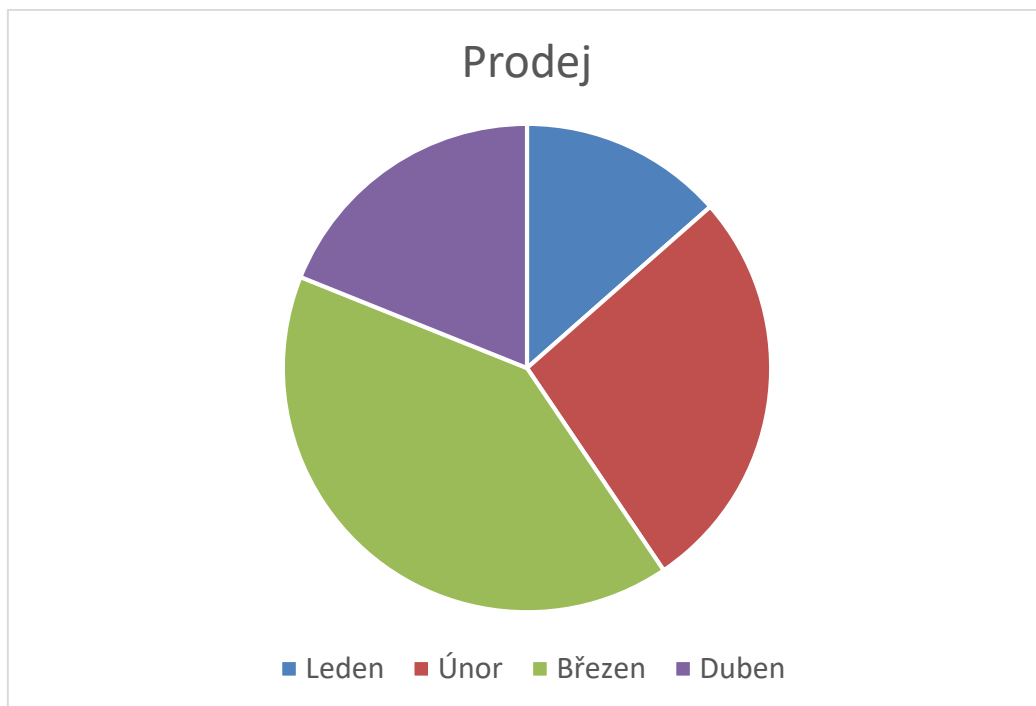


Obr. 3.3: Spojnicový graf zobrazující dvě dimenze a jeden fakt

3. Výsečový graf

Tento typ grafu zobrazuje zkoumanou množinu dat celým kruhem a jednotlivé podmnožiny jako kruhové výseče. Nejčastěji je tento typ grafu využíván pro znázornění rozdělení určitého celku. V multidimenzionálních databázích jej lze využít pro zobrazení jedné dimenze a jednoho či více faktů.

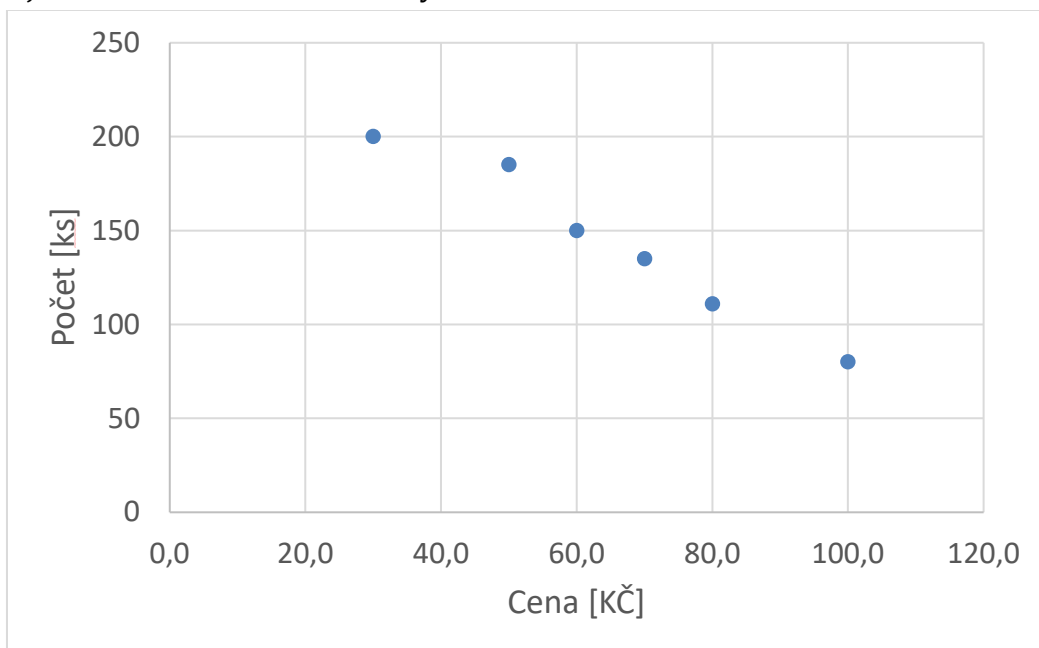
Příkladem budiž množina dat jedné dimenze a jednoho faktu. Lze s výhodou využít jednu z množin uvedených u příkladu výše (příklad spojnicového grafu). Mějme tedy množinu, jež je definována $K = \{5, 10, 15, 7\}$. Jednotlivé hodnoty množiny udávají počet prodejů v uvedených měsících. Výsledný výsečový graf pro tuto množinu hodnot vidíme na obr. 3.4. U tohoto příkladu jsou výseče dostatečně rozdílné a je snadné rozlišit jejich velikosti. V případě, že se některé výseče grafu budou lišit jen nepatrně, je nutné doplnit do grafu i hodnoty jednotlivých kruhových výsečí. Dalším způsobem, jak zajistit rozlišení kruhových výsečí vzhledem k celku, je uvedení procentuální hodnoty dané výseče.



Obr. 3.4: Výšečový graf zobrazující jednu dimenzi a jeden fakt

4. Graf XY

Graf XY, někdy také označován jako bodový graf, zobrazuje jednotlivé hodnoty množiny dat jako body v rovinné kartézské soustavě souřadnic. Lze ji využít ke zkoumání vztahů mezi proměnnými, jež spolu nějakým způsobem souvisí. Tento typ grafu lze využít pro nenulový počet dimenzí a minimálně dva fakty. Tento typ grafu lze vidět na obr. 3.5. Jde o typ grafu s jednou dimenzí a dvěma fakty.



Obr. 3.5: Graf XY zobrazující jednu dimenzi a dva fakty

5 PASIVNÍ OPTICKÉ SÍTĚ (PON)

Jelikož výsledná aplikace bude pracovat s tzv. GPON rámci, je v této kapitole ve stručnosti shrnuta problematika pasivních optických sítí (PON) se zaměřením na gigabitovou pasivní optickou síť (GPON). V případě zájmu o tuto problematiku lze více informací nalézt zde [21], [22].

Pasivní optická síť je optickou přístupovou sítí, jež využívá pouze pasivních přístupových prvků. Optické přístupové sítě (OAN) se skládají z několika funkčních celků a k přenosu dat využívají optická vlákna.

Funkčními celky optických přístupových sítí jsou:

Optické linkové zakončení (OLT)

Tento prvek optické přístupové sítě zajišťuje funkce přístupového rozhraní mezi stranou přístupové sítě a sítí zajišťující telekomunikační služby.

Optická distribuční síť (ODN)

ODN jsou optické přenosové prostředky mezi OLT a ONU.

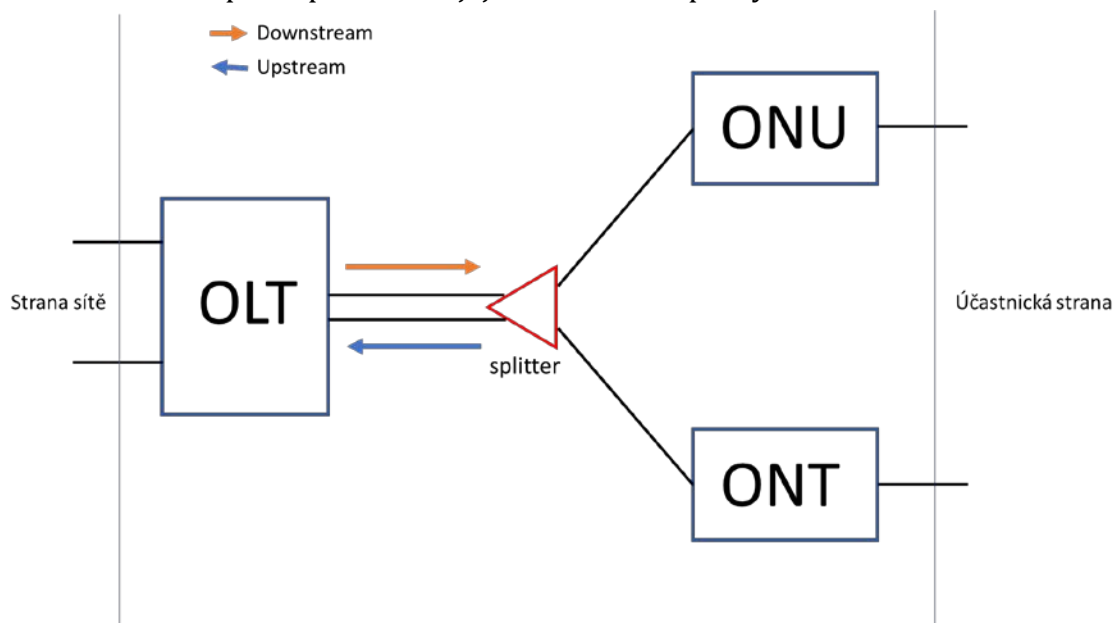
Optické ukončující jednotky (ONU)

Optické ukončující jednotky umožňují použití funkcí rozhraní mezi optickou a metalickou částí přístupové sítě.

Optické ukončující jednotky (ONT)

Tato jednotka zprostředkovává funkce účastnického rozhraní mezi účastnickou stranou (zařízení účastníků) a přístupovou sítí.

Blokové schéma přístupové sítě s jejími funkčními prvky lze vidět na obr. 4.1.



Obr. 4.1: Blokové schéma OAN

Pasivní optická síť využívá také těchto prvků. Optický signál v síti PON je jak již bylo zmíněno realizován pomocí pasivních prvků, rozbočovačů (splitterů). Tyto

rozbočovače slučují účastnické signály, které rozdělí optický signál na nutný počet dílčích směrů bez provádění úprav. Pro obousměrný přenos optického signálu dnes využíváme převážně vlnové dělení WDM (Wavelength Division Multiplexing – vlnový multiplex). Znamená to, že je optický signál přenášen po jednom vlákně a pro Downstream (směr ze sítě k účastníkům) a Upstream (směr od účastníků do sítě) jsou použity rozdílné vlnové délky.

Při vývoji pasivních optických sítí dle přístupů k jejich složení, funkci apod. vznikaly standardy ITU-T a IEEE. Níže je uveden popis části z nich.

ITU-T G.983 (ATM pasivní optická síť – APON)

Tento standard [23] byl prvním standardem pasivních optických sítí. Struktura sítě je zde založena na asynchronním přenosovém režimu (Asynchronous transfer mode – ATM).

Později vznikl standard širokopásmového připojení pasivní optické sítě (BPON), který se vyvinul právě ze standardu APON.

ITU-T G.984 (Gigabitová pasivní optická síť – GPON)

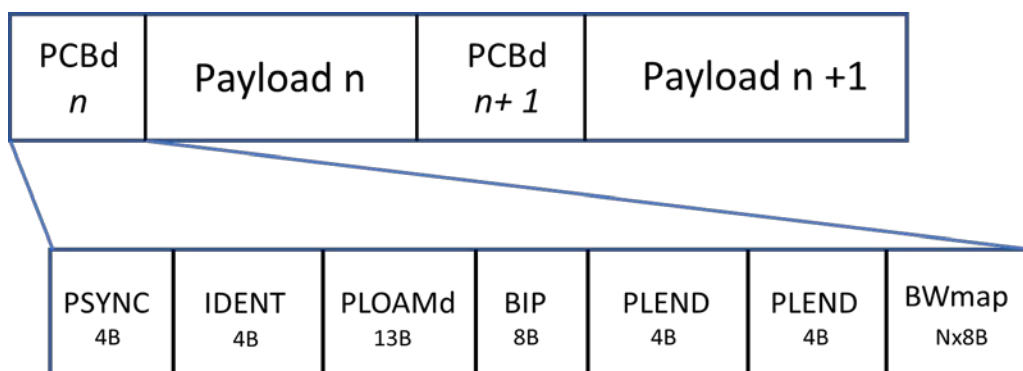
Jde o standard [24], jež se vyvinul z předcházejícího standardu BPON. S použitím tohoto standardu lze přenášet transportní data typu Ethernet a časového multiplexu (TDM) s využitím zapouzdření GPON (GPON encapsulating method – GEM).

Přenos pomocí standardu GPON probíhá za pomoci GEM, jež poskytuje spojově orientovanou komunikaci. Ke komunikaci využívá tzv. GPON rámce pro upstream i downstream.

- **Formát rámce pro downstream**

Provoz pro downstream směřuje od jednotek OLT k jednotkám ONU pomocí časového multiplexu. Každá jednotka ONU přijme pouze rámce, které jsou určeny pro ni.

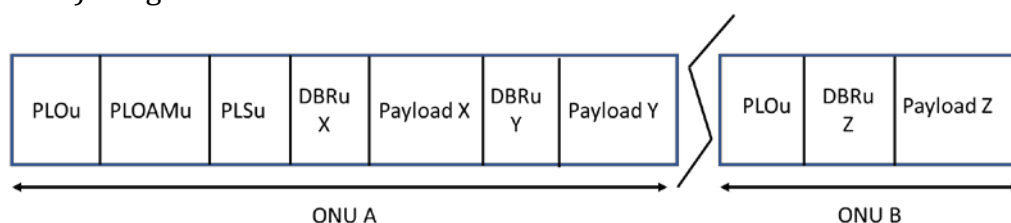
Rámec pro downstream se skládá z kontrolního bloku downstreamu (PCBd) a části GEM. Diagram tohoto rámce lze vidět níže na obr. 4.2.



Obr. 4.2: Diagram rámce pro downstream

- **Formát rámce pro upstream**

Provoz pro upstream využívá časový multiplex s mnohonásobným přístupem (TDMA). Tento provoz směřuje od jednotlivých ONU k jednotce OLT. Rámec pro downstream se skládá z mnoha přenosových dávek (burst). Každá dávka se skládá z horní fyzické vrstvy (physical layer overhead upstream - PLOu), operační, administrativní a řídicí vrstvy (Physical Layer Operations, Administration and Management upstream - PLOAMu), sekvence vyrovnání výkonu (Power Leveling Sequence upstream - PLSu) a zprávy o dynamické šířce pásma (Dynamic Bandwidth Report upstream - DBRu). Diagram rámce lze vidět na obr. 4.3.



Obr. 4.3: Diagram rámce pro upstream

Pasivní optické sítě mohou být využívány pro spojení bod-bod (point to point – p2p) či bod-více bodů (point to multipoint – p2mp). Spojení bod-bod je využíváno pro náročnější zákazníky vyžadující vysoké přenosové rychlosti. Naopak spojení bod-více bodů je využíváno u běžných zákazníků. Jde o levnější řešení, ovšem s nižšími přenosovými rychlostmi než spojení bod-bod.

6 TVORBA ZDROJOVÉ DATABÁZE S DATY GPON

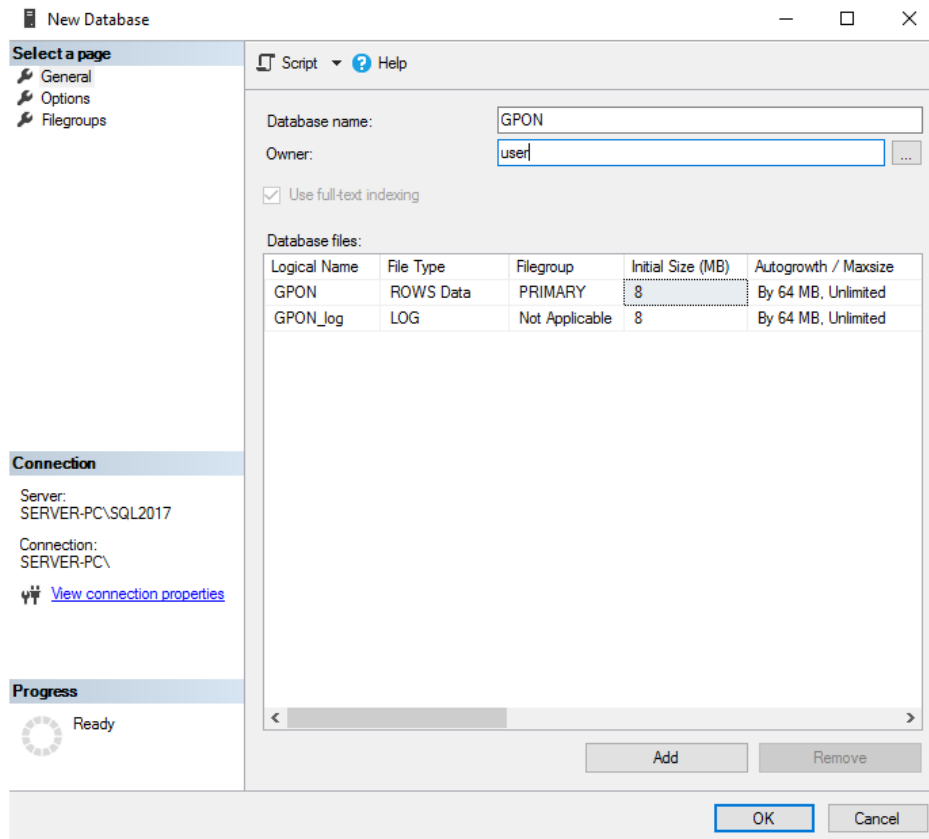
Pro vytvoření multidimenzionální databáze a výsledného zobrazení dat je třeba mít databázi, z níž budou data čerpána. Doba, kdy stačilo zapisovat data ručně na papír či s příchodem prvních počítačů na děrné štítky je již nenávratně pryč. V dnešní době je využíváno k ukládání dat databází. Jako příklad je možno uvést databázi od firmy Microsoft (Microsoft SQL Server) nebo od firmy Oracle (MySQL). Pro výslednou aplikaci byl vybrán Microsoft SQL Server, a to jak pro zdrojovou, tak i pro multidimenzionální databázi.

Zdrojová data, s nimiž pracuje výsledná aplikace, jsou získávána z monitorování sítě GPON a ukládána do databáze Microsoft SQL. Aby mohla být data ukládána, je třeba aby byly před chystány tabulky se správnou strukturou a jejich položky musejí mít správný formát. Níže je popsáno vytvoření databáze a potřebných tabulek v Microsoft SQL Serveru.

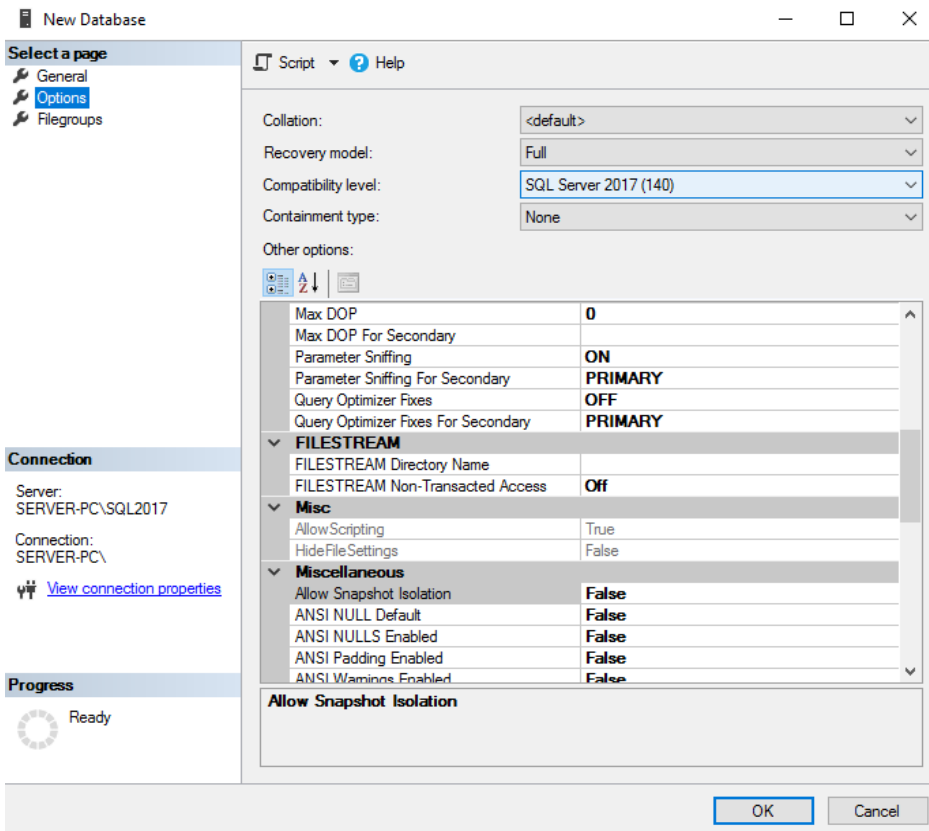
Vytvoření databáze

Nejprve je nutné se přihlásit k Microsoft SQL Serveru, což provedeme pomocí Microsoft SQL Server Management Studia. Po jeho spuštění se zobrazí přihlašovací obrazovka. Pokud je zde nainstalován i analytický server, je potřeba správně zvolit položku *Server type* (typ serveru). Zde lze využít volbu *Database Engine*. Dále je třeba vyplnit jméno serveru a přihlašovací údaje. Po úspěšném přihlášení jsou vlevo zobrazeny položky (složky) serveru.

Databázi vytvoříme kliknutím pravým tlačítkem na složku *Databases* a vybráním *New database*. V nově zobrazeném okně lze nastavit parametry budoucí databáze. V levém horním rohu jsou tři skupiny nastavení: *General* (hlavní), *Options* (nastavení), *Filegroups* (souborové skupiny). Hlavní nastavení (*General*) umožňuje nastavení názvu databáze, majitele databáze (*Owner*) a základních databázových souborů. V další skupině nastavení (*Options*) lze vytvářenou databázi nastavit podrobněji. Je možné zde nastavit například stupeň kompatibility nebo rozsáhlost zálohování. V poslední části nastavení (*Filegroups*) lze vytvořit skupiny souborů k nimž je možné přiřadit jednotlivé databázové soubory. Pro tento příklad databáze stačí vepsat název databáze, případně určit vlastníka a kompatibilitu. Obě nastavení nalezneme na obr. 5.1 a na obr. 5.2. Po nastavení všech požadovaných položek stiskneme *OK*.



Obr. 5.1: Nastavení databáze (General)

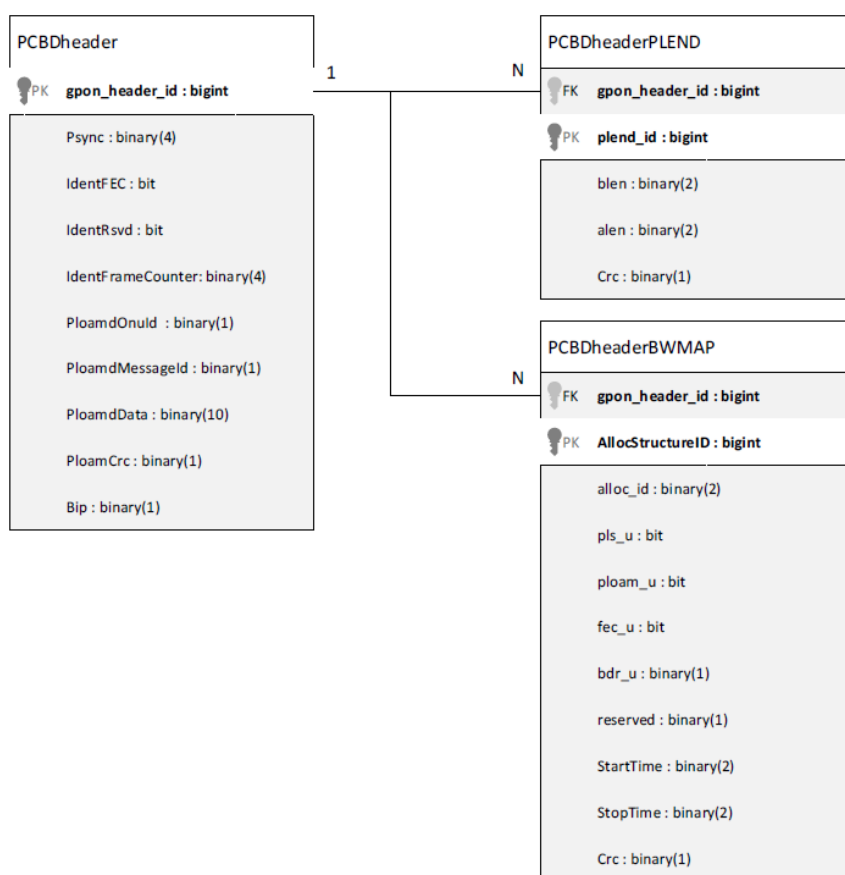


Obr. 5.2: Nastavení databáze (Options)

Nově vytvořenou databázi nalezneme ve složce *Databases* po dvojím kliknutí na tuto složku nebo po kliknutí na znaménko plus vlevo vedle názvu databáze. Stejným způsobem lze rozbalit i nově vytvořenou databázi.

Vytvoření tabulek pro ukládání GPON rámců

Dle struktury dat získaných monitorováním GPON sítě bylo navrženo schéma tří tabulek, jež je možné vidět níže na obr. 5.3. Schéma je tvořeno tabulkami PCBDheader, PCBDheaderPLEND a PCBDheaderBWMAP. Data jsou do tabulek ukládána v uvedeném pořadí. Na tomto schématu vidíme také návaznosti tabulek a jejich primární (primary key - PK) a sekundární (foreign key - FK) klíče (obrázek klíče a zkratka PK, FK). Tabulky jsou propojeny pomocí primárního klíče tabulky PCBDheader, jež je pro druhé dvě tabulky klíčem sekundárním.



Obr. 5.3: Schéma tabulek pro GPON


Nejdůležitější z těchto tří tabulek a také tabulkou využitou pro výsledné zobrazení je tabulka PCBDheader. Její primární klíč se generuje automaticky a je definován typem bigint, aby v případě velkého množství dat nedošlo k vyčerpání primárních klíčů. V pořadí druhým polem (řádkem) tabulky je Psync, do něj je ukládána informace o synchronizační posloupnosti PSYNC. Dle standardu je velikost této posloupnosti 4 bajty, z toho důvodu je typem pole Psync typ binary o délce 4 bajty.

Následující tři pole obsahují informace o poli IDENT. Pole IdentFEC a IdentRsvd jsou datového typu bit. Do pole tohoto typu lze uložit hodnotu pouze o velikosti jednoho bitu. Pole IdentFrameCounter je datového typu binary o délce čtyř bajtů. Do dalších čtyř polí jsou po významných bajtech rozděleny informace o PLOAM zprávách. PloamdOnuId je typu binary o délce jednoho bajtu, jelikož mu náleží jeden bajt zprávy PLOAM. V poli PloamdMessageId je uloženo, o jaký typ PLOAM zprávy se jedná. Tato informace je velikosti jednoho bajtu (typ binary o délce jednoho bajtu). Deset bajtů dat z PLOAM zprávy je uloženo do pole PloamData typu binary o délce deset bajtů. Poslední bajt z PLOAM zprávy je uložen v poli PloamCrc. Tento poslední bajt obsahuje informaci o kontrolním součtu (CRC – Cyclic redundancy check) celé zprávy PLOAM.

Ve druhé tabulce (PCBDheaderPLEND) jsou uchovávány informace o poli PLEND ze záhlaví GPON rámce. Primárním klíčem této tabulky je pole plend_id. Stejně jako v případě primárního klíče u výše jmenované tabulky je typem plend_id bigint. Tato tabulka obsahuje také cizí klíč, jímž je primární klíč tabulky PCBDheader, tedy gpon_header_id. Za těmito klíči jsou pole blen, alen a Crc, jež představují položky BLEN, ALEN a CRC v poli BLEND. Pro uložení těchto dat je využit typ binary. V případě polí blen a alen jde o typ binary o délce dvou bajtů a pole Crc využívá typu binary o délce jednoho bajtu.

Třetí, tedy poslední tabulka ukládá alokované struktury z pole BWMAP. Primárním klíčem je zde pole ALLocStructureID a jeho typem je jako v případě obou předcházejících tabulek bigint. Cizí klíč je totožný s druhou tabulkou, tedy gpon_header_id z tabulky PCBDheader. Dále následuje pole alloc_id typu binary o délce dvou bajtů, jež definuje, pro jakou ONU je alokovaná struktura určena. Následující čtyři pole pls_u, fec_u, bdr_u a reserved jsou definována pro uložení příznaků. Další pole (StartTime a StopTime) definují vysílací čas ONU jednotky. Posledním polem je Crc typu binary o délce jeden bajt pro informace o CRC.

Tabulku lze v databázi vytvořit kliknutím pravého tlačítka na složku *Tables* v patřičné databázi a vybráním *New* a dále *Table*. V novém okně jsou zobrazeny tři sloupce, a to *Column Name* (Název sloupce), *Data Type* (Datový typ) a *Allow Nulls* (povolení/zakázání nulové hodnoty). Příklad vyplnění patřičných hodnot dle schématu výše nalezneme níže na obr. 5.4. Po vyplnění požadovaných hodnot lze tabulku uložit kliknutím na ikonu diskety v horním ovládacím panelu. V otevřeném okně vypíšeme název tabulky a klikneme na tlačítko *OK*. Nově vytvořenou tabulku lze nalézt v patřičné databázi ve složce *Tables*.

SERVER-PC\SQL201...PON - dbo.Table_1* X			
	Column Name	Data Type	Allow Nulls
	gpon_header_id	bigint	<input type="checkbox"/>
	Psync	binary(4)	<input checked="" type="checkbox"/>
	IdentFEC	bit	<input checked="" type="checkbox"/>
	IdentRsvd	bit	<input checked="" type="checkbox"/>
	IdentFrameCounter	binary(4)	<input checked="" type="checkbox"/>
	PloamdOnuld	binary(1)	<input checked="" type="checkbox"/>
	PloamdMessageld	binary(1)	<input checked="" type="checkbox"/>
	PloamData	binary(10)	<input checked="" type="checkbox"/>
	PloamCrc	binary(1)	<input checked="" type="checkbox"/>
	Bip	binary(1)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Obr. 5.4: Schéma tabulky PCBDheader

7 NÁVRH APLIKACE PRO ANALYTICKÉ ZOBRAZENÍ MULTIDIMENZIONÁLNÍCH DAT

Základními předpoklady pro vytvoření aplikace pro analýzu multidimenzionálních dat jsou:

Databáze s daty pro analýzu

- Pro tento projekt je využita databáze s hodnotami vyskytujícími se v hlavičce GPON rámce.

Vytvoření OLAP krychle

- V kapitole 3.5.3 je názorně ukázána tvorba OLAP krychle v programu Microsoft Visual Studio, avšak výsledná aplikace bude využívat pro vytvoření OLAP krychle kódu, který bude popsán níže.

Kód pro načtení a zobrazení požadovaných dat

- Tento kód bude podrobněji popsán níže v kapitole 7.1.1.

7.1 Aplikace pro zobrazení multidimenzionálních dat

Máme-li databázi s daty, je třeba nejprve vytvořit OLAP krychli, a poté je možné začít se samotnou aplikací. Níže je popsáno vytvoření aplikace, po jejímž spuštění bude vytvořena OLAP krychle a zobrazen graf, jehož parametry byly zadány v programu. Pro pochopení uvedených kroků je klíčová znalost předcházející kapitoly 3.5 a základní znalost jazyka C#, který je použit pro aplikaci. Použitým rozhraním pro aplikaci je WPF (Windows Presentation Foundation). Jde o rozhraní pro návrh a zobrazování uživatelského prostředí. Pro graf byl zvolen typ pruhový (sloupcový), jelikož je pro tento případ nejvhodnější.

7.1.1 Tvorba aplikace

Vytvořenou aplikaci je možno nalézt na přiloženém CD. Lze ji otevřít v již zmíněném vývojovém prostředí Microsoft Visual Studio. Jednotlivé soubory (třídy) jsou uvnitř kódu aplikace vhodně okomentovány. Níže jsou uvedeny použité knihovny, vytvoření WPF aplikace a přehled a popis důležitých souborů (tříd).

Knihovny využité v aplikaci

Pro zobrazení grafu je použita knihovna LiveCharts. Její použití se řídí licencí MIT [26]. Instalace této knihovny je provedena pomocí konzole v programu Microsoft Visual Studio. Nejprve je nutno vložit složky s názvem *LiveCharts*, které jsou obsahem přiloženého projektu, umístěny ve složce *packages* (jsou tři), do složky *packages* v projektu, kde bude knihovna LiveCharts použita. Následně v programu

Microsoft Visual Studio v liště nahoře klikneme na tlačítko *Nástroje* dále *Správce balíčku NuGet* a následně *Konzola Správce balíčků*. Konzola je zobrazena ve spodní části okna programu. Do konzole vepíšeme následující příkaz:

```
Install-Package LiveCharts.Wpf
```

Příkaz potvrdíme stiskem klávesy *enter*. Nyní můžeme používat třídy a metody této knihovny. Zde [27] se nachází projekt s ukázkou použití LiveCharts. V případě neúspěchu při instalaci knihovny je třeba zkontrolovat, zda je pro aplikaci použita verze .NET Framework 4.6.1.

Pro připojení k databázi potřebujeme knihovnu AdomdClient. Importuje se stejně jako v předcházejícím případě, změna je pouze v příkazu. Příkaz pro nainstalování knihovny AdomdClient je následující:

```
Install-Package Microsoft.AnalysisServices.AdomdClient
```

Dále je nutné přidat balíček AnalysisServices pro práci s analytickým serverem. Lze jej importovat opět pomocí *Správce balíčku NuGet*. Příkaz pro nainstalování tohoto balíčku je následující:

```
Install-Package Microsoft.AnalysisServices.retail.amd64-Version 15.0.2
```

Tvorba Windows Presentation Foundation aplikace

Výsledná aplikace byla vyvíjena v programu Microsoft Visual Studio. Rozhraním pro vytvoření aplikace byl zvolen WPF [28] (Windows Presentation Foundation), jež je součástí .NET frameworku od verze 3.0. Toto rozhraní je vhodné pro tvorbu grafických aplikací. Níže je popsáno vytvoření projektu aplikace s rozhraním WPF.

V programu Microsoft Visual Studio lze vytvořit projekt s rozhraním WPF kliknutím na volbu *Soubor* vpravo nahoře, výběrem *Nový* a dále *Projekt*. V novém okně lze z levého sloupce vybrat položku *Visual C#* a ze zobrazených nabídek projektů pak *Aplikace WPF (.NET Framework)*. Nyní stačí projekt pojmenovat a kliknout na tlačítko *OK*.

Otevře se nový projekt a je zobrazena jeho první strana reprezentovaná souborem MainWindow.xaml. Lze si všimnout, že část programu (zobrazovací část) je tvořena v jazyce XAML (eXtensible Application Markup Language). Jde o upravený jazyk XML (eXtensible Markup Language), což je značkový jazyk, upravený pro tvorbu aplikací. V průzkumníku řešení lze nalézt všechny soubory vytvořeného projektu. Nový projekt aplikace WPF obsahuje pět souborů, výše uvedený MainWindow.xaml, k němu přidružený MainWindow.xaml.cs, App.xaml, přidružený App.xaml.cs a App.config. První zmíněná dvojice je nejdůležitější, stará se o zobrazení dat a o předání dat k zobrazení. Soubor App.config udává nastavení v projektu například verzi .Net Frameworku. Níže je uveden soupis důležitých souborů obsažených ve výsledné aplikaci, jejímž základem je výše popsáný projekt.

Seznámení s důležitými soubory v projektu

Nejdůležitějšími soubory výsledné aplikace jsou: CubeBuilder.cs, Tools.cs, MainWindow.xaml.cs a MainWindow.xaml. Tyto soubory obsahují metody, které jsou pro běh aplikace klíčové.

CubeBuilder.cs

Tato třída jako celek slouží k vytvoření OLAP krychle. Je vytvořena tak, aby se po svém spuštění připojila k databázi i analytickému serveru a z dat v relační databázi vhodným způsobem vytvořila OLAP krychli. Níže jsou zobrazeny a popsány její metody a klíčové proměnné, jež je nutné správně vyplnit ke korektnímu fungování aplikace.

- **Důležité proměnné**

- **serverName**

- Tato proměnná udává jméno k serveru, ke kterému je nutno se přihlásit.

- **cubeName**

- Jde o proměnnou, jež udává jméno budoucí OLAP databáze.

- **measureTableName**

- Zde je třeba vyplnit název tabulky s fakty.

- **dbName**

- V této proměnné je uložen název zdrojové databáze.

- **cubeDSName**

- Proměnná cubeDSName obsahuje název datového zdroje.

- **cubeDSViewName**

- Název pohledu datového zdroje je uložen v této proměnné.

- **dimensionCount**

- Zde je uložen počet dimenzí výsledné OLAP krychle.

- **tableNameAndKeyAtt**

- Tato proměnná je narození od výše uvedených polem, které obsahuje názvy tabulek s jejich primárními klíči.

- **tableAttributes**

- Stejně jako proměnná tableNameAndKeyAtt je i tato proměnná polem a jsou v ní uloženy názvy atributů budoucích dimenzí.

- **ConnectAS**

- Metoda ConnectAS slouží pro vytvoření připojení k Analytickému serveru. Toto připojení bude uloženo v proměnné s názvem objServer typu *Server*.

- **CreateDatabase**

- Jak již anglický název napovídá, jde o metodu, jež vytvoří databázi. Vytvořená databáze je multidimenzionální databází a nachází se v Analytickém serveru. Nově vytvořená databáze je metodou vrácena v proměnné s názvem objDatabase typu *Database*.

- **DataSourceCreate**

V této metodě je vytvořen datový zdroj. Je důležité zde správně nastavit cestu k databázi tzv. *ConnectionString*. Tento parametr je rozdílný dle serveru, na který se připojuje. Výsledný datový zdroj je vrácen v proměnné s názvem *dataSource* typu *RelationalDataSource*.

- **GenerateDSVSchema**

Po vytvoření datového zdroje lze vytvořit schéma pohledu datového zdroje. Jde o schéma budoucí multidimenzionální kostky, jež bude vytvářena z dat v relační databázi. Tato metoda ke svému fungování využívá metodu *FillDataSet*, která je popsána níže. Výstupem je zde proměnná s názvem *dataSet* typu *DataSet*.

- **FillDataSet**

FillDataSet je pomocnou metodou, která vyčítá data z relační databáze a vhodně vkládá do proměnné *dataset*. Tato proměnná je jejím výsledným výstupem.

- **CreateDataSourceView**

Dalším krokem je vytvoření pohledu datového zdroje. Úkolem této metody je tedy vytvořit za pomoci dříve vytvořeného schématu pohled datového zdroje a zapsat jej do multidimenzionální databáze.

- **CreateDimension**

Posledním krokem před vytvořením samotné OLAP krychle je vytvoření dimenzí, což je, jak název napovídá, úkolem této metody. Ke své funkci využívá metodu *GenerateDimension*, která vytvoří dimenzi dle dostupných dat z datového zdroje. Tato metoda vytvořené dimenze zapíše do proměnné s názvem *dimensions* typu *Dimension[]*, kterou vrátí.

- **GenerateDimension**

Metoda *GenerateDimension* slouží jako pomocná metoda metodě *CreateDimension*. Tvoří jednotlivé dimenze dle dat získaných z dříve vytvořeného datového zdroje. Tyto dimenze pak předá metodě *CreateDimension*.

- **CreateCube**

Nakonec, po vytvoření všech potřebných součástí multidimenzionální databáze, lze přikročit k vytvoření samotné OLAP krychle. Tato metoda tedy vytváří OLAP krychli. K tomuto účelu využívá pomocné metody *GenerateCube*. Po úspěšném ukončení metody *CreateCube* nalezneme v Analytickém serveru novou multidimenzionální databázi a v ní umístěnu nově vytvořenou OLAP krychli.

- **GenerateCube**

GenerateCube je pomocnou metodou pro metodu *CreateCube* a slouží k přidání jednotlivých dimenzí a jejich atributů do nové OLAP krychle.

Tools.cs

Tato třída (soubor) obsahuje metody (viz níže), jež zajišťují data pro zobrazení v grafu.

- **Connector**

Metoda Connector (konektor), jak již název vypovídá, se stará o připojení k databázi. Zároveň je zde také předáván příkaz v jazyce MDX, charakterizující požadovaná data z databáze OLAP.

Návratovou hodnotou této metody je proměnná typu CellSet, ve které je uložena odpověď na dotaz příkazu MDX. Tato návratová hodnota je použita jako vstupní hodnota pro všechny ostatní metody v této třídě.

- **NameOfColumns**

Následující metoda, v překladu názvy sloupců, získává názvy jednotlivých sloupců návratové tabulky, ukládá je a vrací v proměnné typu List.

- **NameOf Rows**

Překlad názvu této metody je názvy řádků. Metoda tedy hledá, ukládá a vrací názvy jednotlivých řádků návratové tabulky v proměnné typu List.

- **Values**

Metoda Values (hodnoty), vyčítá hodnoty návratové tabulky. Jde o hodnoty na spojnici řádků a sloupců.

MainWindow.xaml.cs

V této třídě nalezneme kód naplňující proměnné, které jsou přeneseny do zobrazovaného grafu. Kód v příložené aplikaci je vhodně okomentován, není tedy nutné uvádět jeho podrobnější popis. Nejdůležitějšími proměnnými jsou: SeriesCollection, která uchovává názvy sloupců pomocné tabulky a jeho hodnoty (data), Labels, uchovávající názvy řádků tabulky, jako popisky jednotlivých sloupců výstupního grafu a Formater, jež nastavuje formát popisků v grafu.

MainWindow.xaml

Programový kód tohoto souboru v uvedené aplikaci zajišťuje zobrazení dat do grafu. V případě této ukázkové aplikace jde o graf sloupcový. Jak již koncovka tohoto souboru napovídá, je zde využit jazyk XAML (Extensible Application Markup Language), jde o deklarativní jazyk určený k návrhu uživatelského prostředí v technologii WPF (Extensible Application Markup Language) [29].

Jelikož komentování v tomto typu souboru nezřídka kdy vyvolá chybové hlášení, ve stručnosti bude popsán níže.

V základním tagu (FrameworkElement.Tag [30], jež je základním prvkem v jazyce XAML) s názvem *Window* jsou uvedeny zdroje pro vytvářené zobrazení. Zdrojem pro knihovnu LiveCharts je:

```
xmlns:lvc="clr-namespace:LiveCharts.Wpf;assembly=LiveCharts.Wpf"
```

Dalším důležitým tagem je *Grid*, který charakterizuje vzhled grafu a jeho obsah. V tomto případě jde o graf sloupcový. Lze si dále všimnout klíčového slova *Binding*, které zajišťuje propojení dat v proměnných, které byly naplněny v souboru *MainWindow.xaml.cs*. Jelikož s knihovnou *LiveCharts* je možné vytvářet množství typů grafů nejlépe lze pochopit obsah tohoto tagu z příkladů v projektu uvedených zde [31].

8 ZÁVĚR

Je zřejmé, že informace jsou v současném světě velice důležité a jejich význam stále vzrůstá. Ovlivňují každodenní lidská rozhodnutí, od zvolení správné trasy při cestě do práce, až po rozhodnutí majitelů firem o uzavření správné smlouvy, přesunu výroby či vstupu na nový trh. Čím jsou rozhodnutí důležitější, tím více správných informací vyžadují. Je tedy výhodné vědět, jak je získávat a využívat.

V teoretické části této práce byla uvedena problematika analýzy dat získaných dolováním dat, možnosti jejich uložení a zobrazení. Tato problematika byla rozdělena do několika samostatných problémů, jež byly studovány a rozděleny do jednotlivých kapitol. Jelikož aplikace naprogramovaná v praktické části využívá data získaná monitorováním pasivní optické sítě, konkrétně data z GPON rámců, byla v této práci stručně shrnuta i problematika pasivních optických sítí. Na základě získaných informací byla vytvořena aplikace.

V praktické části byla naprogramována aplikace, jež převádí poskytnutá data z relační databáze do multidimenzionální databáze a provádí jejich zobrazení. Při tvorbě této aplikace bylo nutné vybrat programovací jazyk a programové rozhraní. Programovacím jazykem byl zvolen C#. Rozhraním pro výslednou aplikaci bylo vybráno rozhraní WPF, jež je vhodné pro zobrazování požadovaného typu dat. Důležitým krokem byl také výběr vhodného typu zobrazení (typu grafu). Jako nejvhodnější byl zvolen pruhový (sloupcový) graf. Zdrojová data i multidimenzionální data jsou uložena v databázi Microsoft SQL Server a MS Analysis Server.

Samotné naprogramování výsledné aplikace lze rozdělit do tří částí: vytvoření multidimenzionální databáze a krychle OLAP, získání dat z multidimenzionální databáze a zobrazení získaných dat. Nejnáročnější z těchto tří částí byla část zobrazení dat. Jelikož rozhraní WPF neposkytuje nástroj pro vytvoření grafu, bylo nutné použít již existující knihovnu nebo vytvořit vlastní kód pro zobrazování dat, což by bylo velice časově náročné. Po zvážení dostupných možností byla zvolena volně dostupná knihovna LiveCharts, která umožňuje mnoho způsobů grafického zobrazení dat.

Výsledná aplikace tedy zobrazuje pruhový graf, jenž udává hodnoty dvou nejčastěji se měnících atributů dvou jednotek ONU sítě GPON. Aplikace je připravena pro větší objem dat, než byl k dispozici a případně ji lze pro něj upravit.

Literatura

- [1] Han, J., Kamber, M., Pei, J. Data mining Concepts and Techniques, Elsevier Inc., 2012, 703, ISBN 978-0-12-381479-1
- [2] BERKA, P. Dobývání znalostí z databází. Academia, 2003, 366, ISBN 80-200-1062-9
- [3] MELOUN, M., MILITKÝ, J. Kompendium statistického zpracování dat. Academia, 2002
- [4] Antoch, J., Vorlíčková, D.: Vybrané metody statistické analýzy dat. Academia, Praha, 1992.
- [5] MELOUN, M., MILITKÝ, J., HILL, M. Statistická analýza vícerozměrných dat v příkladech. Karolinum, 2012, ISBN: 978-80-200-2071-0
- [6] Sinčák, P., Andrejková, G. Neurónové siete (Dopredné siete), Elfa-press. 1997, ISBN 80 88786-38-X
- [7] Sinčák, P., Andrejková, G. Neurónové siete (Rekurentné a modulárne neurónové siete), Elfa-press.1997, ISBN 80-88786-42-8
- [8] Darwin, C. R. On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life. London. 1859.
- [9] HYNEK, J. Genetické algoritmy a genetické programování. Praha: Grada,2008. 182, ISBN 978-80-247-2695-3.
- [10] Ballard, Ch., Farrell, D. M., Gupta, A., Mauzuela, C., Vohnik, S. Dimensional Modeling: In a Business Intelligence Environment IBM Corp. [online]. 2006 [cit. 1. 12. 2017]. Dostupné z URL: <<https://www.redbooks.ibm.com/redbooks/pdfs/sg247138.pdf>>
- [11] ISO/IEC 9075-1:2016, Information technology-Database languages-SQL-Part 1: Framework (SQL/Framework), 2016, 94, ISBN 9780580822391
- [12] SIKORA, Z. M. Oracle Database Principles. Palgrave, London, 1997, 275, ISBN: 978 1 349-14693-2
- [13] KRUNTORAD, J. M. History of The CA IDMS Database Management System. IEEE Educational Activities Department Piscataway, 2009, 106, DOI: 10.1109/MAHC.2009.100
- [14] Blasgen, M. W., Astrahan, M. M., Chamberlin, D. D., Gray, J. N., King, W. F., Lindsay, B. G., Lorie, R. A., Mehl, J. W., Price, T. G., Putzolu, G. R., Schkolnick, M., Selinger, P. G., Slutz, D. R., Strong, H. R., Traiger, I. L., Wade, B. W., Yost, R. A. IBM Systems Journal. IBM. 1999, Volume: 38, Issue: 2.3. DOI: 10.1147/sj.382.0375
- [15] Microsoft, MDX Query Fundamentals. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <<https://docs.microsoft.com/en-us/sql/analysis-services/multidimensional-models/mdx/mdx-query-fundamentals-analysis-services>>
- [16] Microsoft, Microsoft Visual Studio Community. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <<https://www.visualstudio.com/cs/free-developer>>

offers/>

- [17] Microsoft, Microsoft Edice systému SQL Express. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <<https://www.microsoft.com/cs-cz/sql-server/sql-server-editions-express>>
- [18] Microsoft, Microsoft Edice systému SQL Developer. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <<https://www.microsoft.com/cs-cz/sql-server/application-development>>
- [19] Microsoft, Microsoft SQL Server Data Tools. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <<https://docs.microsoft.com/en-us/sql/ssdt/download-sql-server-data-tools-ssdt>>
- [20] Microsoft, Microsoft Adventure works DW 2016. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <<https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks>>
- [21] FILKA, M. Optické přenosy informací pro integrovanou výuku VUT a VŠB-TUO. Brno: FEKT VUT v Brně, 2014, 85, ISBN 978-80-214-5064-6
- [22] FILKA, M. Optoelectronics for telecommunications and informatics. Brno: VUT FEEC v Brně, 2009
- [23] ITU-T G.983.1 Broadband optical access systems based on Passive Optical Networks (PON) [online]. 2005 [cit. 10. 5. 2018]. Dostupné z URL: <<https://www.itu.int/rec/T-REC-G.983.1-200501-I/en>>
- [24] ITU-T G.984.1 Gigabit passive optical networks (GPON): General characteristics [online]. 2008 [cit. 10. 5. 2018]. Dostupné z URL: <<http://www.itu.int/rec/T-REC-G.984.1-200803-I/en>>
- [25] IEEE 802.3ah IEEE Standard for Information technology – Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. [online] [cit. 10. 5. 2018]. Dostupné z URL: <http://www.ieee802.org/21/doctree/2006_Meeting_Docs/2006-11_meeting_docs/802.3ah-2004.pdf>.
- [26] LiveCharts Licence, [online] [cit. 10. 5. 2018]. Dostupné z URL: <<https://github.com/beto-rodriguez/Live-Charts/blob/master/LICENSE.TXT>>
- [27] LiveCharts Zdrojový kód, [online] [cit. 10. 5. 2018]. Dostupné z URL: <<https://github.com/beto-rodriguez/Live-Charts>>
- [28] Stephens, R. WPF Programmer's Reference: Windows Presentation Foundation with C# 2010 and .NET 4, 2010, ISBN: 978-0-470-62473-9
- [29] Microsoft, What is XAML?. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <<https://msdn.microsoft.com/en-us/library/cc295302.aspx>>
- [30] Microsoft, FrameworkElement.Tag Property. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <[https://msdn.microsoft.com/en-us/library/system.windows.frameworkelement.tag\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.frameworkelement.tag(v=vs.110).aspx)>

- [31] LiveCharts Basics, [online] [cit. 10. 5. 2018]. Dostupné z URL: <<https://lvcharts.net/App/examples/v1/wpf/Basics>>
- [32] LACKO, L. Databáze: Datové sklady analýza OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle, Computer Press, 2003, 483, ISBN 80-7226-969-0
- [33] MAIMON, O., ROKACH, L. Data mining and knowledge discovery handbook, second edition, Springer Science + Business Media, 2005, 1306, ISBN 978-0-387-09822-7
- [34] Microsoft, Data Mining. [online]. 2017 [cit. 10. 5. 2018]. Dostupné z URL: <[https://msdn.microsoft.com/en-us/library/ms174949\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/ms174949(v=sql.120).aspx)>
- [35] ZENDULKA, J., BARTÍK, V., LUKÁŠ, R. Získávání znalostí z databází – studijní opora. Brno: FIT VUT v Brně, 2006.
- [36] TVRDÍK, J. Analýza vícerozměrných dat, Ostravská univerzita, 2013, 128
- [37] KUŽELA, A. Vizualizace dat. [online] [cit. 10. 5. 2018]. Dostupné z URL: <<http://www.fit.vutbr.cz/study/courses/VPD/public/0708VPD-Kuzela.pdf>>
- [38] Cale, I., Salihovic. A., Ivekovic. M. Gigabit Passive Optical Network – GPON, IEEE, 2007, ISBN: 953-7138-09-7

Seznam symbolů, veličin a zkratek

APON	ATM Passive Optical Network
ATM	Asynchronous transfer mode
BPON	Broadband Passive Optical Network
EPON	Ethernet Passive Optical Network
GPON	Gigabit Passive Optical Network
BI	Business Intelligence
CRC	Cyclic Redundancy Check
GPON	Gigabit Passive Optical Network
KDD	Knowledge discovery in databases
MDX	Multidimensional Expressions
MS	Microsoft
OAN	Optical Access Network
ODN	Optical Distribution Network
OLAP	Online Analytical Processing
OLT	Optical Line Termination
ONT	Optical Network Terminal
ONU	Optical Network Unit
PON	Passive Optical Network
P2P	Point To Point
P2MP	Point To Multipoint
QBE	Query By Example
SQL	Structured Query Language
SSDT	SQL Server data tools
TDM	Time Division Multiplex
WDM	Wavelength Division Multiplexing
WPF	Windows Presentation Foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

Seznam příloh

Příloha 1. DVD s výslednou aplikací